

Elektronisen kauppapaikan toteuttaminen liikkuvilla agenteilla

Harri Huntus

Tampereen yliopisto
Tietojenkäsittelyopin laitos
Pro gradu -tutkielma
Lokakuu 1999

Tampereen yliopisto

Tietojenkäsittelyopin laitos

Harri Huntus: Elektronisen kauppapaikan toteuttaminen liikkuvilla agenteilla

Pro gradu -tutkielma, 72 sivua

Lokakuu 1999

Tässä tutkimuksessa keskitytään elektronisiin kauppapaikkoihin ja niiden toteuttamiseen liikkuvilla agenteilla. Käsiteltäessä elektronisia kauppapaikkoja keskitytään elektronisten kauppapaikkojen tarkoitukseen, niistä saataviin hyötyihin, niiden keskeisiin ongelmiin ja ongelmien erilaisiin ratkaisuihin.

Liikkuvia agentteja käsiteltäessä vertaillaan liikkuvia agentteja paikallaan pysyviin agentteihin ja pohditaan liikkuvien agenttien hyötyjä ja ongelmia. Lisäksi pohditaan, miten liikkuvia agentteja voitaisiin käyttää hyödyksi ratkaistessa elektronisten kauppapaikkojen ongelmia.

Liikkuvien agenttien kehittämiseen tarkoitetuista kehitysympäristöistä tutustutaan muutamaa parhaiten dokumentoituun. Näistä on valittu IBM:n kehittämä Aglets-järjestelmä, jonka avulla toteutetaan liikkuviin agentteihin pohjautuva elektroninen kauppapaikka. Toteutettava järjestelmä on tarkoitettu yksityishenkilöille yksittäisten tavaroiden kauppaamiseen eli kyseessä on lähinnä eräänlainen elektroninen kirpputori.

Tutkielman lopussa arvioidaan valittua ratkaisua ja pohditaan, kuinka elektroniset kauppapaikat voisivat kehittyä tulevaisuudessa.

Avainsanat ja -sanonnat: agentti, liikkuvat agentit, aglet, elektroninen kauppapaikka, elektroninen kaupankäynti.

Sisällys

1.	Johdanto.....	1
2.	Elektroninen kauppapaikka.....	4
2.1.	Kauppapaikkojen tarkoitus.....	4
2.2.	Elektronisesta kauppapaikasta saatavia hyötyjä.....	5
2.3.	Toimivia elektronisia kauppapaikkoja.....	6
2.4.	Elektronisten kauppapaikkojen ongelmia.....	6
2.4.1.	Toiseksi tekeytyminen.....	7
2.4.2.	Välimies-tekniikka.....	8
2.4.3.	Tietoturva	8
2.4.4.	Resurssien valvonta	9
2.4.5.	Käytön valvonta	9
2.4.6.	Maksaminen.....	10
2.5.	Yhteenveto.....	12
3.	Agentit elektronisessa kaupankäynnissä.....	13
3.1.	Agenttiluokittelu	14
3.2.	Kauppa-agentit	14
3.3.	Käyttäjän tunnistaminen agenttipohjaisessa järjestelmässä.....	15
3.4.	Käyttäjän tukeminen elektronisen kaupankäynnin eri vaiheissa	15
3.5.	Neuvottelu.....	18
3.6.	Agenttipohjaisen järjestelmän käytettävyys.....	18
3.7.	Agenteilla toteutettuja elektronisia kauppapaikkoja	19
3.7.1.	Kasbah.....	20
3.7.2.	e-Marketplace.....	21
3.8.	Yhteenveto.....	22
4.	Liikkuvat agentit.....	24
4.1.	Itsenäisen liikkuvan agentin määritelmä.....	24
4.2.	Agenttien tärkeitä ominaisuuksia.....	25
4.3.	Liikkuvien agenttien vertailu paikallaan pysyviin agentteihin	27
4.4.	Agenttien liikkuvuudesta saatavia hyötyjä.....	28
4.4.1.	Verkon kuorman väheneminen.....	28
4.4.2.	Riippumattomuus protokollista.....	29
4.4.3.	Laajemmat oikeudet.....	29
4.4.4.	Palvelujen saavutettavuus	29
4.4.5.	Reagoiminen virhetilanteisiin.....	30
4.4.6.	Hajautettujen järjestelmien ohjaus	31
4.5.	Agenttien liikkuvuuden ongelmia.....	31

4.5.1.	Yhtenäisen järjestelmän tarve	31
4.5.2.	Järjestelmän vaatima ylläpito	32
4.5.3.	Järjestelmien monimutkaistuminen	32
4.5.4.	Agenttien villiintyminen	33
4.6.	Kehitysympäristöjä liikkuvien agenttien toteuttamiseen	33
4.6.1.	MASIF	34
4.6.2.	Java-To-Go	35
4.6.3.	Mole	35
4.6.4.	Concordia	36
4.6.5.	Grasshopper	37
4.6.6.	JATLite	38
4.6.7.	Kehitysympäristöjen vertailua	39
4.6.8.	Muita kehitysympäristöjä	40
4.7.	Yhteenveto	41
5.	Agletit	42
5.1.	Agenttien vertailua appletteihin	42
5.2.	Agenttipalvelin	43
5.3.	Agenttien ympäristö	44
5.4.	Agenttien elinkaari	45
5.5.	Agentin käyttäytyminen	47
5.5.1.	Luonti ja tuhoaminen	47
5.5.2.	Siirtyminen	48
5.5.3.	Kommunikointi	49
5.6.	Agenteista saatava hyöty elektronisille kauppapaikoille	50
5.6.1.	Turvallisuus	51
5.6.2.	Resurssien valvonta	51
5.6.3.	Käytön valvonta	52
5.6.4.	Maksaminen	52
5.7.	Yhteenveto	53
6.	Oma järjestelmä	54
6.1.	Arkkitehtuuri	55
6.2.	Järjestelmässä tarvittavia agentteja	58
6.2.1.	Hallinnointiagentit	58
6.2.2.	Kauppaa käyvät agentit	61
6.2.3.	Muut agentit ja oliot	63
6.3.	Järjestelmän käyttöliittymä	63
6.4.	Toteutuksen arvointia ja vertailua	65
7.	Pohdintaa	67
8.	Yhteenveto	71

1. Johdanto

Jo kauan on ennustettu, että tulevaisuudessa tietoverkko tulee olemaan lähes jokaisessa taloudessa. Viime vuosina kehitys onkin edennyt vahvasti tähän suuntaan, etenkin *World Wide Web:n* (WWW) suuren suosion ansiosta. Tulevaisuudessa tietoverkon käyttöönotto tulee olemaan entistä vaivattomampaa. Enää ei tarvittaisi erillistä verkkoa pelkästään tietokoneiden kommunikointia varten, vaan kaikkea dataliikennettä varten olisi yksi yhteinen dataverkko.

Tulevaisuuden dataverkossa liikkuisivat niin tv- ja radiolähetykset kuin muukin tietoliikenne. Koska dataverkon kautta välitettäisiin esimerkiksi tv-lähetyksiä, tultaisiin valtaosa talouksista liittämään tähän dataverkkoon. Samalla yhä useammat taloudet saataisiin liitettyä kiinteästi tietoverkkoon entistä helpommin. Tätä tietoverkkoa voitaisiin käyttää myös langattomasti matkapuhelinten välityksellä lähes mistä käsin tahansa. Verkon yleistyessä ja sähköisten palvelujen kehittyessä verkosta tulee yhä arkipäiväisempi osa ihmisten elämää.

Verkossa tullaan hoitamaan jokapäiväisiä askareita aina ruokaostoksia myöten. Jo jonkin aikaa kuluttajat ovat voineet hankkia verkon välityksellä erilaisia tuotteita (esimerkiksi kirjoja, levyjä ja juuri ruokatuotteita), mutta etenkin tulevaisuudessa erilaiset elektroniset kauppapaikat tulevat olemaan erittäin keskeisessä asemassa ihmisten elämässä.

Verkkoon tulee yhä laajemmin saataville myös erilaisia viranomaispalveluita [Espoo, 1999], joita käyttämällä käyttäjä voi välttyä turhauttavalta jonottamiselta. Vastaavasti myös suurta osaa pankkipalveluista voi jo nykyisin käyttää verkon kautta. Verkosta kehittyy siis entistä hyödyllisempi paitsi kaupankäynnissä niin myös virallisten asioiden hoitamisessa ja näin ollen verkosta tulee yhä arkipäiväisempi väline käyttäjälle.

Viranomaisten kiinnostus tietoverkkoa kohtaan auttaa myös elektronisten kauppapaikkojen leviämistä, sillä viranomaiset joutuvat ratkaisemaan samoja ongelmia kuin elektronisten kauppapaikkojen ylläpitäjät. Esimerkiksi viranomaisten luodessa kaikille suomalaisille sähköisen tunnisteen [Väestörekisterikeskus, 1999a], on tästä hyötyä myös elektroniselle kaupalle asiakkaan tunnistamisessa.

Perinteisen kaupan tapaan elektronisessa kaupankäynnissä tarvitaan kauppiaalta kuluttajalle -tyyppisten kauppapaikkojen lisäksi myös kauppiaalta kauppiaille ja kuluttajalta kuluttajalle -tyyppisiä kauppapaikkoja muun

muassa tukkuporrasta ja käytettyjen tavaroiden markkinoita varten. Käyttäjällä tulisi olla verkossa käytettävissä kaikki sellaiset kaupan muodot, joita käyttäjä voi käyttää myös perinteisessä kaupassa.

Elektronisen kauppapaikan tai kauppapaikan toimintaa tukevan palvelun toteuttamisessa voidaan käyttää hyödyksi myös agentteja. Useimmiten agentteja käytetään tiedon suodatukseen tai tiedonhakuun, mutta erityisesti liikkuvien agenttien avulla agenttien käyttötarkoitusta voidaan laajentaa. Käsitteenä liikkuvat agentit on tunnettu jo useita vuosia [Harrison *et al*, 1995], mutta liikkuvat agentit eivät ole tulleet käytännössä erityisen suosituiksi. Liikkuvilla agenteilla ja moniagenttijärjestelmillä voidaan kuitenkin ratkaista monimutkaisiakin ongelmia, joten niiden käyttö lisääntynee lähivuosina.

Kaupankäynti on siirtymässä yhä suuremmassa määrin tietoverkkoihin, jonka myötä elektronisten kauppapaikkojen merkitys jokapäiväisessä elämässä kasvaa entisestään. Elektronisten kauppapaikkojen muodostuessa yhä tärkeämmiksi on tärkeää, että niitä voidaan käyttää mitä erilaisemmissa tilanteissa. Näin ollen pitää myös kehittää uusia tapoja elektronisten kauppapaikkojen saavutettavuuden parantamiseksi. Liikkuvia agentteja voidaan hyödyntää myös elektronisten kauppapaikkojen toteutuksessa. Tässä tutkimuksessa keskitytään elektronisiin kauppapaikkoihin ja niiden toteuttamiseen liikkuvilla agenteilla. Liikkuvat agentit valittiin kauppapaikan toteuttamistavaksi siksi, että niiden avulla voidaan kohtuullisella vaivalla kehittää laajojakin järjestelmiä monimutkaistenkin ongelmien ratkaisemiseksi. Kehittämällä muutamia yhteistoiminnallisia agentteja voidaan luoda jopa maailmanlaajuinen järjestelmä, jossa agentteja luodaan eri puolilla maailmaa ja lähetetään ne suorittamaan tehtävänsä tietyille palvelimelle.

Luvussa kaksi käsitellään elektronisia kauppapaikkoja keskittyen muutamiiin jo olemassa oleviin ratkaisuihin, elektronisten kauppapaikkojen keskeisiin ongelmiin (muun muassa turvallisuus, resurssien valvonta) ja ongelmien erilaisiin ratkaisuihin.

Luvussa kolme pohditaan miten elektroninen kaupankäynti voi hyötyä agenteista ja esitellään muutama agenteilla toteutettu elektroninen kauppapaikka.

Luvussa neljä käsitellään liikkuvia agentteja. Liikkuvien agenttien kohdalla keskitytään liikkuvien agenttien yleisiin toimintaperiaatteisiin sekä verrataan liikkuvia agentteja paikallaan pysyviin agentteihin. Luvussa viisi keskitytään erityisesti IBM:n kehittämään Aglets-järjestelmään [IBM, 1999a].

Tämän pohjalta on kehitetty oma, Aglets-järjestelmään perustuva kokeellinen elektroninen kauppapaikka. Toteutettu kauppapaikka on suunnattu lähinnä yksityishenkilöiden käyttöön ja yksittäisten tavaroiden kauppaamiseen

– ratkaisu on siis lähinnä eräänlainen elektroninen kirpputori. Kauppapaikan toteutusta on käsitelty tarkemmin luvussa kuusi.

Lopuksi käsitellään muutamia elektronisten kauppapaikkojen ja liikkuvien agenttien ongelmallisia osa-alueita elektronisten kauppapaikkojen toteuttamisen suhteen sekä mahdollisia uusia kehitettäviä ratkaisuita.

2. Elektroninen kauppapaikka

Elektroniseksi kauppapaikaksi voidaan yleisesti ottaen ymmärtää sähköisesti toteutettu kauppajärjestelmä, jossa yksi tai useampi myyjä käy kauppaa yhden tai useamman ostajan kanssa. Yksi esimerkki tällaisesta kauppapaikasta on elektroninen kirjakauppa Amazon [1999].

Ensimmäiset suuremman suosion saaneet elektroniset kauppapaikat myivät pääsääntöisesti kirjoja, cd-levyjä tai muita vastaavia viihdetuotteita. Tällä hetkellä elektronisia kauppapaikkoja on lukuisia ja niistä on saatavissa myös jokapäiväisiä hyödykkeitä. Esimerkiksi Helsingissä toimiva Ruok@.net [1999] toimittaa ruokatavaroita pääkaupunkiseudulla ja myös perinteisenä ruokakauppana toimiva Ruokavarasto [1999] usealla paikkakunnalla.

Päivittäistavaroita ja muita hyödykkeitä kuluttajille kauppaavat elektroniset kauppapaikat tulevat lisääntymään entisestään tietoverkkojen käyttäjäkunnan kasvaessa. Myös kynnys uusien elektronisten kauppapaikkojen perustamiseen on entistä pienempi, koska tietoverkkojen käyttäjät ovat jo omaksuneet elektronisten kauppapaikkojen käyttämisen aikaisemmin perustettujen kauppapaikkojen ansiosta.

2.1. Kauppapaikkojen tarkoitus

Elektronisten kauppapaikkojen tarkoituksena on yleensä täyttää yksi tai useampia seuraavista tavoitteista: (1) potentiaalisten kauppakumppaneiden tunnistaminen, (2) tietyn kauppakumppanin valinta tai (3) varsinaisen kaupankäynnin suorittaminen [Chouldhury *et al*, 1998]. Elektroninen kauppapaikka voi siis olla tarkoitettu muuhunkin kuin vain kaupankäynnin suorittamiseen; elektroninen kauppapaikka voi olla kehitetty vain kauppakumppaneiden tunnistamista tai valitsemista varten.

Yllä mainitut esimerkit (Amazon, Ruok@.net ja Ruokavarasto) on kehitetty pelkästään kaupankäynnin suorittamiseen; ne eivät millään tavoin tue kauppakumppanin tunnistamista tai valintaa, sillä niiden elektronisilla kauppapaikoilla myydään vain yhden kauppakumppanin tuotteita. Käyttäjän suorittaessa ostoksiaan jollakin em. kauppapaikasta on kauppakumppani siis tällöin jo valittu. Jos kyseessä olisi esimerkiksi elektroninen ostoskeskus, jossa on useampia saman alan yrityksiä, olisi myös kauppakumppanin valinta tuettuna.

Elektroniseksi ostoskeskukseksi voidaan kutsua paikkaa, johon on kerätty useita eri elektronisia kauppapaikkoja samaan yhteyteen. Periaatteessa kuka tahansa voi perustaa eräänlaisen elektronisen ostoskeskuksen keräämällä linkkejä eri elektronisiin kauppapaikkoihin. Ostoskeskus voi tukea myös kauppakumppanin valintaa, jos ostoskeskuksessa on esillä useampi kuin yksi saman alan elektroninen kauppapaikka. Kehittyneempi tapa tukea kauppakumppanin valintaa olisi jonkin kauppakumppanin suositteluasiakkaan sijainnin tai ostomieltyymysten perusteella.

Yksi esimerkki elektronisesta ostoskeskuksesta on tv-mainoksillakin tunnettavuuttaan lisännyt Meritan Solotori [Merita, 1999]. Solotorin elektroniseen ostoskeskukseen on valittu sellaisia kauppapaikkoja, jotka käyttävät maksamiseen Meritan elektronista Solo-maksua. Tietokonemyyjät Netissä -palveluun [Salminen, 1999] puolestaan on kerätty vain atk-alan laitteita myyvien liikkeiden elektronisia kauppapaikkoja tai WWW-sivuja.

Nykyisissä elektronisissa kauppajärjestelmissä jää käyttäjän vastuulle yksi kaupankäynnin peruselementeistä: tarpeen huomaaminen. Verkossa toimivat elektroniset kauppapaikat eivät voi huomata käyttäjän tarpeita. Tarpeen huomaaminen on kuitenkin vaativa tehtävä ja se jäänee käyttäjän vastuulle myös tulevaisuudessa lukuunottamatta helposti ennustettavissa olevia hankintoja.

2.2. Elektronisesta kauppapaikasta saatavia hyötyjä

Elektroniseksi kauppapaikaksi voitiin ymmärtää tietojärjestelmä, joka yhdistää useita ostajia ja myyjiä. Laajassa järjestelmässä ostajan tarvitsee siis olla yhteydessä vain yhteen kauppapaikkaan, jotta hänellä olisi saatavissa tietoa useiden myyjien kauppaamista tuotteista [Chouldhury *et al*, 1998]. Vastaavasti myyjän tarvitsee olla yhteydessä vain yhteen kauppapaikkaan, jotta hänen tuotteillaan olisi suuri potentiaalinen ostajakunta.

Laajasta elektronisesta kauppapaikasta voi siis olla hyötyä sekä myyjälle että ostajalle. Myyjän ei tarvitse etsiä mahdollisia jakelukanavia, jolloin myös mahdolliset väliportaot myyjän ja ostajan välillä poistuvat. Näin ollen myyjä hyötyy voidessaan tarjota ostajalle tuotteitaan alempaan hintaan. Mahdollisesti myös pienemmät vuokra- ja muut kulut ovat myyjälle hyödyksi: koska ostaja ostaa myyjän tuotteita elektronisen kauppapaikan kautta, ei myyjän tarvitse maksaa korkeaa vuokraa sijainniltaan parhaasta myyntipaikasta, vaan myyjä voi toimia lähes mistä tahansa.

Ostajan taasen on helpompi vertailla eri myyjien tuotteita keskenään, jos useita myyjiä tarjoaa tuotteitaan samassa kauppapaikassa. Ostaja voi hyötyä myös myyjän kulujen laskemisesta tuotteiden alempina hintoina.

2.3. Toimivia elektronisia kauppapaikkoja

Nykyään on jo olemassa lukuisia erilaisia elektronisia kauppapaikkoja tai kauppajärjestelmiä. Suomessakin voi jo nykyisistäkin kauppapaikoista ostaa lähes mitä tahansa. Akateeminen Kirjakauppa [1999] myy kirjoja, kuten myös ehkä Internetin tunnetuin elektroninen kauppapaikka, amerikkalainen Amazon [1999]. Anttilan nettikaupasta [Anttila, 1999] voi tilata muun muassa cd-levyjä, astioita ja retkeilytavaroita. Cd- ja dvd-levyjä voi tilata myös esimerkiksi Boxman:sta [1999].

Myös päivittäiset ruokaostokset onnistuvat elektronisten kauppapaikkojen kautta. Pääkaupunkiseudulla asuva voi jopa valita tilaako ruokansa vain pääkaupunkiseudulla toimivasta Ruok@.net:stä [1999] vai useammalla paikallakunnalla toimivasta Ruokavarastosta [1999]. Yksi ensimmäisistä ruokatavaroita Internetissä kauppaava yritys oli amerikkalainen Peapod [1999].

Lippupalvelun [1999] FiLippus nettipalvelun kautta käyttäjä voi ostaa lippuja Lippupalvelun välittämiin viihde-, urheilu- ja kulttuuritapahtumiin.

Lisäksi on lukuisia palveluita, joiden kautta käyttäjä voi tutustua myytävänä oleviin kohteisiin, mutta ei välttämättä tilata itse tuotetta. Yksi tällainen palvelu on esimerkiksi Vehon [1999] vaihtoautomyynti, jonka kautta käyttäjä voi etsiä Vehossa myynnissä olevia vaihtoautoja haluamillaan kriteereillä.

Elektronisten kauppapaikkojen kautta voi siis hankkia jo tälläkin hetkellä mitä erilaisempia tavaroita tai palveluita. Tulevaisuudessa tarjonta kasvaa entisestään.

2.4. Elektronisten kauppapaikkojen ongelmia

Elektroniset kauppapaikat ovat suhteellisen uusi ilmiö, joten myös kauppapaikkojen toiminta- ja toteutustavat ovat suhteellisen uusia. Tästä johtuen elektronisissa kauppapaikoissa on vielä paljon kehitettävää. Parannettavaa löytyy muun muassa turvallisuudesta: koska elektronisten kauppapaikkojen yhteydessä voi liikkua salauksen vaativaa tietoa (esimerkiksi luottokorttien numeroita, henkilötietoja, salasanoja) on kauppapaikkojen kohdalla kiinnitettävä huomiota järjestelmän turvallisuuteen. Turvallisuutta voidaan yrittää rikkoa monella tavalla, esimerkiksi salakuuntelemalla toisen henkilön viestejä tai tekeytymällä toiseksi henkilöksi. Muita keskeisiä ongelmia on esimerkiksi resurssien valvonta, käyttäjän tunnistaminen ja ostoksien maksaminen.

Elektronisen kauppapaikan turvallisuus voi vaarantua monella tapaa. Paitsi että itse järjestelmän turvallisuus voi vaarantua, on vaarana myös, että käyttäjien henkilötietoja vuotaa järjestelmän ulkopuolelle. Vaikkeivat käyttäjien henkilötiedot sinänsä salaisia olisikaan, ovat käyttäjien sähköpostiosoitteet, käyttäjäprofiilit ynnä muut käyttäjästä jotain kertovat yksityiskohdat nykypäivänä arvokasta tietoa. Esimerkiksi mainostajat ovat kiinnostuneita käyttäjien tiedoista, koska niiden avulla he pystyvät kohdentamaan mainoksiaan paremmin. Elektronisen kauppajärjestelmän pitääkin huolehtia paitsi itse kaupankäynnin turvallisuudesta myös siitä, ettei yksittäisen käyttäjän tietoihin päästä käsiksi. Seuraavassa on lyhyesti kuvattu muutamia elektronisten järjestelmien yleisesti tunnettuja turvallisuusongelmia sekä tekniikoita päästä käsiksi arkaluontoisiin tietoihin.

2.4.1. Toiseksi tekeytyminen

Ehkä helpoin tapa vaarantaa lähes minkä tahansa järjestelmän turvallisuus on toiseksi henkilöksi tekeytyminen. Näin voidaan saada selville esimerkiksi jonkun muun henkilön luottokortin numero tai muuta arkaluontoista tietoa. Kuten kaikkien elektronisten järjestelmien myös elektronisten kauppapaikkojen kohdalla riskinä ovat myös ohjelmat, jotka jäljittelevät alkuperäisten ohjelmien toimintaa saadakseen selville käyttäjän käyttäjätunnuksen ja salasanan tai muuta arkaluontoista tietoa.

Usein tällaiset ohjelmat näyttävät ulkoisesti hyvin samanlaisilta kuin oikeakin ohjelma, mutta käyttäjätunnuksen salasanan saatuaan ohjelma ilmoittaa virheestä ja lopettaa toimintansa. Ennen toimintansa lopettamista ohjelma voi kuitenkin lähettää kaappaamansa käyttäjätunnuksen ja salasanan esimerkiksi sähköpostilla tai tallentaa ne levyille. Näin käyttäytyvästä ohjelmasta käytetään yleisesti termiä troijan hevonen.

Toiseksi tekeytymällä aiheutetaan useimmiten harmia lähinnä järjestelmän käyttäjille, joiden salasanoja tai luottokortin numeroita käyttämällä voidaan suorittaa esimerkiksi ostoksia toisen henkilön nimissä; itse järjestelmälle aiheutetaan harvemmin haittaa. Käyttäjien silmissä järjestelmän luotettavuus kuitenkin heikkenee kaikenlaisissa järjestelmään liittyvissä tietoturvaongelmissa.

2.4.2. Välimies-tekniikka

Välimies-tekniikan perusidea on se, että kahden henkilön keskustellessa kolmas henkilö asettautuu heidän väliinsä ja välittää keskustelun henkilöltä toiselle. Näin ollen kolmas osapuoli saa käyttöönsä myös kaiken mahdollisesti salaisenkin materiaalin.

Vaikka keskustelevat henkilöt suojaisivatkin keskusteluaan vaihtamalla keskustelun alussa esimerkiksi salausjärjestelmän julkisia avaimia ja käyttämällä viestien salausta, voi kolmas osapuoli päästä käsiksi tietoihin korvaamalla osapuolien lähettämät julkiset avaimet omalla julkisella avaimellaan. Tällä tekniikalla voidaan ohjelmallisesti puuttua sekä kahden henkilön että kahden ohjelman väliseen keskusteluun. Kolmas osapuoli ei välttämättä muuta välitettyjä viestejä millään tavalla vaan ne voidaan esimerkiksi tallentaa myöhempää käyttöä varten.

2.4.3. Tietoturva

Laajassa järjestelmässä voi olla tuhansia, mahdollisesti jopa satoja tuhansia käyttäjiä. Jokaisesta käyttäjästä on tallennettu henkilötiedot ja mahdollisesti tietoa käyttäjän kiinnostuksen kohteista. Etenkin kuluttajalta kuluttajalle –tyyppisessä järjestelmässä on tärkeää, että kauppaa käyvät osapuolet tietävät jotain toisistaan. Toisaalta taas käyttäjät haluavat, ettei heidän tietojaan pääse leviämään liian laajalle.

Tietoturvan kannalta olisikin tärkeää kyetä löytämään sopiva kompromissiratkaisu: käyttäjän yhteystiedot voi paljastaa esimerkiksi vasta sen jälkeen, kun sopiva ostokohde on löytynyt ja kumpikin osapuoli on suostunut neuvoteltuun hintaan. Kuluttajalta kuluttajalle -tyyppisessä kaupankäynnissä myyjän tai ostajan esiintyminen täysin anonyyminä tuskin onnistuu, koska myyjä ja ostaja todennäköisesti kuitenkin kohtaavat tavaraa luovutettaessa.

Koska käyttäjien tietojen tietoturvallisuus on elektronisen kauppapaikan luotettavuuden kannalta tärkeää, on selvää, että kauppapaikan ylläpito ei saa perustua esimerkiksi käyttäjien tietojen myymiseen mainostajille. Useimpien tällä hetkellä toimivien elektronisten kauppapaikkojen ylläpitäjänä on jokin kauppaketju, joka käyttää käyttäjien yhteystietoja omiin tarkoituksiinsa, esimerkiksi mainostamiseen.

2.4.4. Resurssien valvonta

Järjestelmän ylläpitäjällä tulisi olla mahdollista rajoittaa järjestelmän käyttämiä resursseja. Jos järjestelmä paisuu erittäin suureksi, on vaarana, että järjestelmä kuluttaa niin paljon esimerkiksi prosessointitehoa, että palvelimessa toimivat muut järjestelmät vaarantuvat. Prosessointitehon lisäksi muita tärkeitä resursseja ovat esimerkiksi muisti ja tallennuskapasiteetti. Järjestelmän pitäisi pystyä palvelemaan mahdollisimman monia käyttäjiä ja samalla kuluttaa vähän resursseja. Useille samanaikaisille käyttäjille tarkoitetun järjestelmän tulisikin olla mahdollisimman kevyt, jotta palvelimen ylläpitäjä ei joutuisi rajoittamaan käyttäjien määrää resurssipulan vuoksi.

Resurssien valvonta on tärkeää myös käytettävyyssyistä. Hidas järjestelmä menettää helposti käyttäjien luottamuksen vaikkakin Internetissä hitaus johtuukin useimmiten tietoverkon kuormituksesta. Resurssien valvonnalla ja käyttäjämäärän rajoittamisella järjestelmä pystytään pitämään nopeana. Rajoittamisen huonona puolena on se, että käyttäjä voi saada järjestelmältä ilmoituksen, että järjestelmään ei juuri nyt pääse, koska sitä käyttää niin monta käyttäjää. Tällöin käyttäjä helposti käyttää jotakin toista kauppapaikkaa tai perinteistä kauppaa.

Resurssien valvonta on useissa järjestelmissä kuitenkin tarpeen, sillä muutamat tavat arkaluontoisen materiaalin saamiseksi palvelimelta perustuvat palvelimen ylikuormittamiseen.

2.4.5. Käytön valvonta

Miten käyttäjä tunnistetaan? Järjestelmässä, joka ei vaadi salasanoja, henkilö voi esiintyä jonkin toisen henkilön nimellä. Esimerkiksi useimmissa päivittäistavaroita myyvissä elektronisissa kauppapaikoissa on mahdollista tilata toimituksia toisen henkilön nimellä toisen henkilön osoitteeseen.

Useimmissa sähköpostipalveluissa on käytössä käyttäjän henkilöllisyyden varmistus siten, että käyttäjälle lähetetään sähköpostia käyttäjän ilmoittamaan sähköpostiosoitteeseen ja näin varmistetaan, että käyttäjä on ilmoittanut oman sähköpostiosoitteensa. Vastaavanlainen varmistus voisi toimia myös elektronisten kauppapaikkojen kohdalla: käyttäjälle lähetettäisiin postia hänen ilmoittamaan toimitusosoitteeseen ja näin varmistettaisiin, että hän on ilmoittanut oman osoitteensa. Käyttäjän tilaukset toimitettaisiin sitten aina tähän osoitteeseen. Tällaisen järjestelyn haittapuolena on se, että käyttäjä ei pääse

suorittamaan ostoksia ennen kuin hänen ilmoittamansa osoite on varmistettu postitse.

Useilla kauppaketjuilla on tälläkin hetkellä käytössään erilaisia kanta-asiakasjärjestelmiä, joihin usein liittyy kanta-asiakaskortti. Myös tätä korttia voidaan käyttää käyttäjän tunnistamiseen, koska asiakkaan henkilötiedot on yhdistettävissä hänen käyttämäänsä korttiin. Ostokset voi myös maksaa luottolisella kanta-asiakaskortilla luottokortin tapaan, tällainen järjestelmä on esimerkiksi K-Supermarket Ykköshallilla [1999].

Tulevaisuudessa käyttäjän tunnistamiseen voitaisiin käyttää myös esimerkiksi sähköistä henkilökorttia tai elektronista allekirjoitusta. Erilaisia *henkilön sähköinen tunnistaminen* (HST) kokeiluja on Suomessakin jo useita: esimerkiksi Väestörekisterikeskus ja Suomen Posti Oy käyttivät 23.11.1998 tekemässään sopimuksessa dokumentin sähköistä allekirjoitusta [Lahti, 1999]. Osuuspankkikeskus ja Väestörekisterikeskus puolestaan kokeilevat sähköisen henkilökortin käyttöönottoa Osuuspankin internet-pankissa [MikroPC, 1999]. Ottamalla laajasti käyttöön sähköiset henkilökortit tai allekirjoitukset voitaisiin niitä käyttää henkilön tunnistamiseen salasanojen tai tunnuslukujen sijaan. HST-hankkeesta vastaa Väestörekisterikeskus, jonka lehdistötiedotteen [Väestörekisterikeskus, 1999b] mukaan sähköistä henkilökorttia voidaan anoa 1.12.1999 alkaen. Lehdistötiedotteessa mainitaan myös, että korttia voidaan käyttää julkishallinnon palveluiden lisäksi myös sähköisen kaupankäynnin kehittämiseen.

Sähköinen henkilökortti yhdessä tunnusluvun kanssa on henkilön sähköisen tunnistamisen perusta. Vastaavasti kuin käyttäjätunnuksien ja salasanojen myös sähköisen henkilökortin väärinkäyttö on mahdollista. Sähköinen henkilökortti on kuitenkin luotettavampi kuin salasanoihin ja käyttäjätunnuksiin perustuva tunnistaminen. Jos henkilön salasana ja käyttäjätunnus joutuisivat sivullisten tietoon, voisi näitä tietoja väärinkäyttää useampikin henkilö. Sähköisen henkilökortin ja sitä vastaavan tunnusluvun joutuessa sivullisten tietoon voisi tietoja väärinkäyttää vain se henkilö, jonka hallussa kortti kulloinkin on. Sähköisen henkilökortin luotettavuutta parantaa myös se, että kadotettuaan kortin henkilö voi ilmoittaa kortin sulkulistalle, jonka jälkeen korttia ei voida enää käyttää [Väestörekisterikeskus, 1999a].

2.4.6. Maksaminen

Maksamisessa tulee kiinnittää huomiota sekä maksutapahtuman turvallisuuteen myös itse tapahtuman helppouteen. Käyttäjä haluaa tietää luottokorttinsa numeron liikkuvan verkon yli turvallisesti, mutta erittäin tärkeää on myös se, että käyttäjä ei kohtaa erityisiä ongelmia maksaessaan ostoksia. Kauppiaan

kannalta tilanne on erittäin huono silloin, jos käyttäjä päättää jättää ostokset suorittamatta sen takia, että maksaminen vaikuttaa hankalalta. Mahdollisuuksien mukaan maksamisesta olisi tehtävä käyttäjälle mahdollisimman läpinäkyvää.

Maksaminen tulisi yhdistää mahdollisimman tiiviisti itse ostostapahtumaan. Internetin kautta suoritettavissa ostoksissa maksamattomien laskujen määrä on korkeampi kuin normaalissa kaupassa. Esimerkiksi päivittäistavaroiden ostossa maksujärjestelyiden tulisi olla mahdollisimman nopeita, koska ostokset mahdollisesti toimitetaan jo muutaman tunnin kuluessa.

Vaikka päivittäistavaroiden kohdalla laskun maksaminen toimituksen yhteydessä onnistuukin, ei se kuitenkaan ole ratkaisu kaikille elektronisille kauppapaikoille. Elektronisen kauppapaikan toimipiste voi sijaita myös jossakin toisessa maassa, jolloin laskun toimittaminen toimituksen mukana ei ole enää järkevä ratkaisu. Samassa maassa toimivasta elektronisesta kauppapaikasta tilattujen ostoksien maksaminen toimituksen yhteydessä tulevalla laskulla on yksinkertainen tapa hoitaa maksaminen, mutta tämä tapa ei luontevasti sovi muuten elektronisesti suoritettavaan kauppatapahtumaan.

Parempia tapoja olisi esimerkiksi asiakkaan tilin veloittaminen tai luottokortilla maksaminen. Yli maiden rajojen toimivat elektroniset kauppapaikat suosivatkin luottokortilla maksamista, koska tällöin mahdollisesti maksamattoman laskun periminen ei kuulu kauppapaikalle vaan paikalliselle luottoyhtiölle. Kaikki luottoyhtiöt eivät vielä kuitenkaan hyväksy, että heidän luottokorteillansa suoritetaan ostoksia Internetissä. Luottoyhtiöt pelkäävät, että korttien numerot joutuvat väärin käsiin. Esimerkiksi suomalainen Luottokunta on suostunut luottokorttitietojensa välittämiseen Internetissä vasta syksystä 1999 alkaen [MikroPC, 1999]. Luottokortilla maksaminenkaan ei siis sovi kaikille, eivätkä kaikki edes omista luottokorttia.

Luottokortin ohella helpoimpia tapoja maksaa lasku Internetissä suorite-
tuista ostoksista lienee verkkomaksupalvelu. Verkkomaksupalvelussa käyttäjä ostotapahtuman yhteydessä täyttää tai saa nähtäväkseen esitäytetyn normaalin tilisiirtolomakkeen kaltaisen sähköisen tilisiirtolomakkeen ja maksaa ostokset hyväksymällä tuon tilisiirron. Tällainen maksaminen kuitenkin vaatii sopimuksen elektronisen kauppapaikan ja pankkiyhtiön välillä sekä mikropalvelusopimuksen käyttäjän ja pankin välillä. Itse maksaminen kuitenkin on verkkomaksupalvelua käyttäneelle erittäin helppoa ja maksu rekisteröityy myyjälle välittömästi.

Pelkästään kanta-asiakasjärjestelmään perustuvassa elektronisessa kauppapaikassa myös maksaminen olisi helposti yhdistettävissä esimerkiksi luotolliseen kanta-asiakaskorttiin. Käyttäjällä voi olla käytössään myös elektroninen

lompakko, jota välittäjä veloittaa ja toimittaa maksun elektroniselle kauppapaikalle.

Vaikka maksamiseen onkin olemassa lukuisia erilaisia vaihtoehtoja, tuskin mikään niistä sopii kaikille tai kaikenlaisiin tilanteisiin. Elektronisen kauppapaikan tulisikin tarjota mahdollisimman laaja valikoima erilaisia maksutapoja, joista käyttäjä voisi valita mieleisensä ja tilanteeseen sopivan.

2.5. Yhteenveto

Elektronisella kauppapaikalla tulisi olla jokin selkeä tehtävä, esimerkiksi potentiaalisten kauppakumppaneiden tunnistaminen. Useimpien tällä hetkellä toimivien elektronisten kauppapaikkojen tehtävä on selkeästi varsinaisen kaupan käynnin suorittaminen. Koska elektroniset kauppapaikat ovat suhteellisen uusi ilmiö, liittyy niihin vielä useita ongelmia.

Ehkä keskeisin elektronisiin kauppapaikkoihin liittyvä ongelma-alue on ostosten maksaminen. Tällä hetkellä maksamiseen on olemassa useita eri vaihtoehtoja, mutta tarjolla ei ole sellaista maksutapaa, mikä voisi sopia kaikille käyttäjille. Koska maksamisen yhteydessä liikkuu arkaluonteista tietoa, on kiinnitettävä erityistä huomiota maksamisen turvallisuuteen. Maksamisen olisi myös oltava käyttäjälle mahdollisimman helppoa ja läpinäkyvää.

3. Agentit elektronisessa kaupankäynnissä

Agentilla tarkoitetaan yleisesti ottaen ohjelmaa, joka hoitaa käyttäjän sille määräämää tehtävää. Yleensä yksittäisten agenttien hoitamat tehtävät ovat varsin yksinkertaisia ja rutiininomaisia, mutta jos useita agentteja työskentelee saman ongelman ratkaisemiseksi, voidaan tällaisilla moniagenttijärjestelmillä ratkaista monimutkaisiakin ongelmia.

Staattisella agentilla viitataan agenttiin, joka ei voi siirtyä pois siltä koneelta, jossa se on luotu. Staattinen eli ei-liikkuva agentti on sijoittunut käyttäjän koneelle, josta käsin se suorittaa tehtävänsä. Yleisesti tällaisista agenteista käytetään vain termiä agentti, mutta jotta ero liikkuviin agentteihin olisi selvä, käytetään tässä tutkielmassa termiä staattinen agentti. Termillä agentti taas viitataan kumpaankin edellä mainittuun agenttityyppiin. Jos staattinen agentti tarvitsee jotakin tietoa joltain toiselta koneelta, ottaa staattinen agentti yhteyden tähän koneeseen, noutaa tiedon ja prosessoi tiedon käyttäjän koneella.

Liikkuva agentti puolestaan voi siirtyä käyttäjän koneelta myös jollekin toiselle koneelle. Jos liikkuva agentti tarvitsee jotakin tietoa, joka on jollakin toisella koneella, voi liikkuva agentti siirtyä käyttäjän koneelta toiselle koneelle, prosessoida tiedon siellä ja palata käyttäjän koneelle tuloksien kanssa.

Vaikka agenteista voikin olla hyötyä elektronisessa kaupankäynnissä, kaikki agentit eivät elektroniseen kaupankäyntiin luonnollisestikaan sovellu. Agenteista pitää tunnistaa sellaisia ominaisuuksia, jotka voivat olla hyödyksi nimenomaan elektronisessa kaupankäynnissä.

Agenteista voi olla hyötyä monessa eri elektronisen kaupankäynnin vaiheessa sekä käyttäjän, elektronisen kaupan ylläpitäjän että elektronisen kaupan toteuttajan näkökulmasta. Agentit voivat esimerkiksi tukea käyttäjää kaupankäynnin eri vaiheissa, auttaa elektronisten kauppapaikkojen ongelmien ratkaisemisessa ja tehdä kauppapaikoista käyttäjäystävällisempiä. Tässä kappaleessa käsitellään muun muassa edellä mainittuja asioita. Lisäksi agenteilla voidaan toteuttaa joitakin kauppapaikan toimintaa tukevia palveluita, joiden tarkoituksena on esimerkiksi houkutella kuluttajia tutustumaan kauppapaikkaan. Kappaleen lopussa esitellään muutamia agenteilla toteutettuja elektronisia kauppapaikkoja.

3.1. Agenttiluokittelu

Schubert *et al* [1998] esittelee tavan ryhmitellä agenttijärjestelmiä. Heidän tekemänsä ryhmittely jakaa agenttijärjestelmät kolmiulotteiseen matriisiin perustuen kolmeen agenttien ominaisuuteen: älykkyyteen, liikkuvuuteen ja agenttien määrään (onko agentteja yksi vai monta).

Matriisista he muodostavat agenttien ominaisuuksien perusteella kolme erilaista luokkaa, jotka soveltuvat erilaisiin tehtäviin: informaatioagentit, yhteistyöagentit ja kauppa-agentit. Näistä erityisesti kauppa-agentit soveltuvat käytettäväksi elektronisessa kaupassa.

Vaikka informaatio- ja yhteistyöagentit eivät suoraan soveltuisikaan elektroniseen kauppaan, voitaisiin näitä käyttää hyödyksi myös elektronisten kauppapaikkojen yhteydessä. Informaatio- ja yhteistyöagenttien avulla voidaan toteuttaa esimerkiksi jokin kaupankäyntiä tukeva palvelu tai ostajia houkutteleva viihteellinen palvelu.

3.2. Kauppa-agentit

Schubert *et al* [1998] kuvaavat kauppa-agentteja älykkyydeltään yksinkertaisiksi, liikkuviksi tai staattisiksi agentteiksi, jotka toimivat joko yksin tai moniagenttijärjestelmässä. Kauppa-agenttien tehtäviä voisivat olla muiden muassa tilausten käsittely, elektroninen maksaminen, turvallisuus ja neuvottelut. Yksittäisiä sovellusalueita ovat esimerkiksi elektroninen kaupankäynti ja tuotanto.

Kauppa-agenttien toiminnot ovat usein samankaltaisia informaatioagenttien kanssa. Kauppa-agentit kuitenkin eroavat informaatioagenteista siten, että kauppa-agentteihin on usein lisätty jokin tietty piirre, joka tukee varsinaista kaupankäyntiprosessia. Kauppa-agentit voivat esimerkiksi neuvotella muiden agenttien kanssa käyttäjän määrittämien ehtojen mukaan, tehdä tilauksia tai hoitaa maksujärjestelyjä. Tällaiset kauppa-agentit luovat ideaalisen pohjan agenttiperustaisen elektronisen kauppapaikan luomiselle. Agenttiperustaisessa elektronisessa kauppapaikassa agentit edustavat kaikkia kaupankäynnin osapuolia (myyjä, ostaja, pankki, jne).

Schubert *et al* [1998] mainitsevat kauppa-agenttien tehtäviksi neljä tehtävää: tilausten käsittelyn, elektronisen maksamisen, turvallisuuden ja neuvottelut. Kaikki nämä ovat tärkeitä ominaisuuksia elektronisessa kauppapaikassa. Kauppa-agentit soveltuvatkin elektronisen kauppapaikan toteuttamiseen hyvin. Täytyy kuitenkin muistaa, että vaikka agentit hoitaisivat suurimman

osan kauppaprosessista, on lopullisen päätäntävallan oltava käyttäjällä, käyttäjän niin halutessa.

3.3. Käyttäjän tunnistaminen agenttipohjaisessa järjestelmässä

Yksi erittäin tärkeä osa elektronisessa kauppapaikassa on käyttäjän tunnistaminen. Ilman luotettavaa käyttäjän tunnistamista ei voida olla varmoja tilauksen tehneen käyttäjän henkilöllisyydestä.

Käyttäjä voidaan tunnistaa agenttien avulla varmemmin kuin esimerkiksi sähköpostiosoitteen avulla. Ilmoittautuessaan johonkin palveluun tai esimerkiksi pyytäessään tarjousta jostakin tuotteesta tai palvelusta sähköpostitse, käyttäjä voi helposti ilmoittaa sähköpostiosoitteeseen jonkin muun henkilön sähköpostiosoitteen. Jos järjestelmä perustuu liikkuviin agentteihin, on käyttäjän tunnistaminen varmempaa. Agentti voidaan ohjelmoida niin, että se tietää mikä on sen kotipalvelin (agentti voi tallentaa muistiin sen palvelimen osoitteen, jossa se luotiin). Näin ollen järjestelmä saa luotettavan tiedon pyynnön alkuperästä. Myös mahdollinen vastaus menee oikeaan osoitteeseen, kun vastaus ensin kerrotaan agentille ja sitten agentti määrätään palaamaan kotipalvelimelleen.

Joissakin agenttijärjestelmissä jokaiseen käyttäjän luomaan agenttiin kirjaetaan agentin luojaan sähköpostiosoite tai jokin muu tunniste. Näin ollen jokainen käyttäjän luoma agentti voidaan tunnistaa hänen omakseen ja näin saada agentin alkuperä selville.

3.4. Käyttäjän tukeminen elektronisen kaupankäynnin eri vaiheissa

Kuluttajan käyttäytymismallista kaupankäyntitilanteessa on olemassa useita eri teorioita ja malleja. Guttman *et al* [1998] ovat erotelleet näistä malleista kuusi eri vaihetta, jotka ovat yhteisiä kaikille malleille. Näin kaupankäynti on voitu jakaa kuuteen eri vaiheeseen:

1. *Tarpeen tunnistaminen* on vaihe, jossa kuluttaja tulee tietoiseksi ostotarpeesta. Tarpeen heräämistä voidaan edesauttaa esimerkiksi mainonnalla.
2. *Ostokohteen valinta* on vaihe, jossa kuluttaja kerää tietoa, jotta voisi päättää, mitä hän ostaa. Vaihe voi sisältää esimerkiksi vaihtoehtojen tuotteiden vertailua.

3. *Myyjän valinta* on vaihe, jossa kuluttaja päättää, miltä kauppiaalta hän tuotteensa ostaa. Valintaan vaikuttavia tekijöitä voivat olla muun muassa hinta, takuu, saatavuus ja niin edelleen.
4. *Neuvottelu* on vaihe, jossa päätetään tuotteen lopullinen hinta ja muut kaupantekoon vaikuttavat tekijät. Tämä vaihe voidaan myös ohittaa, sillä esimerkiksi päivittäistavaroita ostaessa hinta ja muut neuvottelun kohteet ovat ennalta määrättyjä, eikä neuvottelun varaa ole.
5. *Osto ja toimitus* on vaihe, jossa kuluttaja on yleensä päättänyt ostaa tietyn tuotteen, jonka jälkeen tuote toimitetaan kuluttajalle.
6. *Tuotetuki ja arviointi* on ostotapahtuman jälkeinen vaihe, jossa kuluttaja käyttää hankkimaansa tuotetta ja arvioi, onko hän tyytyväinen tuotteeseen ja koko kaupankäyntikokemukseen. Vaihe voi sisältää esimerkiksi käyttäjän avustamista hänen tuotteen kanssa kohtaamisissaan ongelmassa.

Parhaiten agentit pystyvät tukemaan käyttäjää tämän mallin vaiheissa 2, 3 ja 4. Vaiheita erotellessa täytyy kuitenkin muistaa, että malli on yksinkertaistus monimutkaisesta käyttäytymismallista, joka voi vaihdella kuluttajasta ja kaupankäyntimuodosta riippuen.

Vaikka tarpeen tunnistaminen yleensä onkin vaihe, joka on täysin riippuvainen kuluttajasta, voidaan kuitenkin myös tätä vaihetta tukea jollakin tavalla. Mikroprosessoreiden yhä yleistyessä agenteja voisi olla mahdollista sijoittaa esimerkiksi jääkaappiin tai omakotitalon öljysäiliöön. Agenteilla olisi näin ollen mahdollista tunnistaa käyttäjän tarpeet etukäteen etenkin päivittäistavaroissa. Käyttäjän ei välttämättä tarvitse seurata esimerkiksi jääkaapista löytyvien ruokien määrää, koska agentti voisi hoitaa tämänkaltaisen rutiinitehtävän käyttäjän puolesta. Agentti voisi myös automaattisesti tilata elektronisesta kaupapaikasta uuden tuotteen loppuvan tilalle. Vastaavasti agentti voisi myös tarkkailla, milloin käyttäjän lempiartistilta tulee uusi levy myytäväksi elektroniseen levykauppaan ja ehdottaa käyttäjälle sen ostamista.

Agenteista voisi olla hyötyä tällaisten yksinkertaisten, säännöllisesti toistuvien tai tarkasti yksilöitävien tarpeiden huomaamisessa, mutta monimutkaisemmissa tapauksissa niistä ei juurikaan ole hyötyä. Agentin on mahdotonta huomata esimerkiksi milloin käyttäjän tarvitsee hankkia uusia vaatteita.

Agentit voivat tukea käyttäjää ostokohteen valinnassa hankkimalla tietoa myytävänä olevista tuotteista käyttäjän haluamassa kategoriassa. Esimerkiksi WWW:stä löytyy runsaasti tietoa erilaisista tuotteitteista, lisäksi elektroniset kaupapaikat voivat tarjota tietoa myymistään tuotteista. Käyttäen hyväksi

näitä tietolähteitä agentti voi tarjota käyttäjälle riittävän määrän tietoa, jotta käyttäjä voi tehdä päätöksen, minkä tuotteen hän ostaa.

Vastaavasti voidaan tukea myös kauppakumppanin valintaa. Agentti voi vertailla kuinka paljon käyttäjän haluama tuote missäkin kauppapaikassa maksaa ja suositella sen (ja muiden saatavilla olevien kriteereiden) perusteella käyttäjälle jotakin kauppapaikkaa. Käyttäjällä on tietysti oltava mahdollisuus nähdä agentin päätökseen vaikuttaneet tiedot ja mahdollisesti valita jokin toinen kauppakumppani agentin suositteleman sijaan.

Agentit voivat tukea käyttäjää myös neuvoteltaessa tuotteen hinnasta tai toimitusehdoista. Aiheesta kerrotaan tarkemmin kohdassa 3.5.

Vaiheiden viisi ja kuusi sisältö riippuu hyvin paljon ostetusta tuotteesta ja kauppakumppanista. Vaikka vaiheiden toteuttaminen olisikin teknisesti mahdollista, kaikkien tuotteiden kohdalla se ei ole järkevää. Esimerkiksi ohjelmistotuotteen kohdalla palautteen antaminen valmistajalle agentin välityksellä on hyvä ratkaisu, mutta samanlainen ratkaisu tuskin toimii esimerkiksi leivän valmistuksessa tapahtuneen virheen korjaamiseksi.

Nakamuran ja Yamamoton [1997] mukaan vaiheet viisi ja kuusi hoidetaan yleensä teollisten standardien mukaan, jotta saavutettaisiin riittävä taso turvallisuudessa, yksityisyydessä ja luotettavuudessa. Esimerkiksi *SET* (Secure Transaction Protocol) tarjoaa protokollan maksamiseen liittyvissä tapahtumissa. *OTP* (Open Trading Protocol) puolestaan pyrkii kattamaan kaiken vaiheisiin viisi ja kuusi sisältyvän toiminnallisuuden.

Järjestelmien edelleen kehittyessä voidaan mahdollisesti myös maksujärjestelyt toteuttaa agenteilla. Elektronisen kaupan yhteydessä on esitelty muun muassa elektroniset lompakot, jotka sijaitsevat käyttäjän omalla koneella. Koska agentti toimii sekä omalla koneella että verkossa, on sillä siis mahdollisuus kommunikoida sekä käyttäjän elektronisen lompakon että kauppapaikan kanssa. Näin ollen agentti voisi siis veloittaa tuotteen hinnan käyttäjän elektronisesta lompakosta ja toimittaa maksun verkossa sijaitsevalle palveluntarjoajalle.

Vaikka etenkin päivittäistavaroiden kohdalla agentit pystyisivät suorittamaan koko ostosprosessin käyttäjän puolesta, ei vielä kuitenkaan liene järkevää antaa kaikkea agentin hoidettavaksi: käyttäjä on kuitenkin vastuussa nimissään toimivan agentin toimista, joten käyttäjän tulisi aina tehdä lopullinen päätös. Vasta kun agentit ovat kehittyneet riittävästi huomaamaan myös hetkittäisetkin muutokset käyttäjän tarpeissa, voidaan niille antaa valta toimia aivan itsenäisesti. Ennen kuin koko ostosprosessi voitaisiin uskoa agenttien hoidettavaksi, pitäisi myös erilaisten teknisten ratkaisuiden laatua ja luotettavuutta pystyä parantamaan: käyttäjä tuskin haluaa ostotapahtuman yhteydessä virheilmoitusta "lompakossasi on tapahtunut sovellusvirhe, ota yhteys laitetoimittajaan".

3.5. Neuvottelu

Neuvottelu on yksi tärkeimmistä osa-alueista myös elektronisessa kaupankäynnissä, sovitaanhan siinä tuotteen lopullisesta hinnasta sekä mahdollisista toimitusehdoista. Neuvottelun tuloksena esimerkiksi tuotteen hinta voi muuttua kymmeniäkin prosentteja. Vaikka päivittäistavaroissa tai pientavaroissa (cd-levyt ynnä muut sellaiset) ei yleensä ole tapana hinnoista neuvotella, tulevat neuvottelut olemaan osa elektronista kauppaa sellaisilla kaupan aloilla, joilla on normaalissakin kaupankäynnissä totuttu neuvottelemaan. Sandholmin [1999] mukaan automaattisen neuvottelun lupaavimpia sovellusalueita ovat elektroninen kaupankäynti, sähkömarkkinat, tuotannon suunnittelu ja tietoliikennesurssien varaus.

Agentit voivat olla käyttäjälle korvaamaton apu neuvottelussa: agentti yrittää väsymättä neuvotella tuotteen hintaa tai toimitusehtoja käyttäjän haluamiseksi eikä luovuta ennen kuin on saavuttanut tason, johon käyttäjä on tyytyväinen. Kahden agentin neuvotellessa tälle tasolle ei aina välttämättä päästä, jos neuvotteluiden lähtötasot ja käyttäjien ehdot ovat liian kaukana toisistaan. Tosin näin voi aivan yhtä hyvin käydä myös neuvotellessa ilman agentteja.

Sandholm [1999] esittelee yhden esimerkin kustannusperusteisesta metodista, jolla agentit neuvottelevat esimerkiksi jonkin palvelun hinnasta. Metodi on varsin yksinkertainen: agentti ottaa hoitaakseen tehtävän, jos sille maksetaan siitä enemmän kuin sen omat kustannukset ovat. Jos agentti löytää jonkin toisen agentin, joka tekee saman tehtävän halvemmalla kuin sen omat kustannukset olisivat, antaa agentti tehtävän tuon halvemman agentin tehtäväksi.

Myös neuvottelun suhteen agenttien laatu ja luotettavuus ovat tärkeitä. Neuvottelun tuloksen on oltava sitova, jotta kauppa voitaisiin tehdä. Ostaja voi vielä neuvottelunkin jälkeen jättää tuotteen ostamatta, mutta kauppiaan on myytävä tuote neuvotteluissa sovitulla hinnalla. Neuvottelevissa agenteissa ei siis saa olla ohjelmointivirheitä, koska ne saattaisivat tuottaa kauppiaille tappiota. Neuvottelevat agentit eivät myöskään saisi villiintyä tai toimia muuten väärin, koska asiakkaan silmissä se laskee myyvän osapuolen luotettavuutta.

3.6. Agenttipohjaisen järjestelmän käytettävyys

Käyttäessään Internetissä sijaitsevaa liikkuviin agentteihin pohjautuvaa elektronista kauppapaikkaa käyttäjä käyttääkin varsinaisesti omalla koneellaan

sijaitsevaa agenttia, joka sitten siirtyy elektroniseen kauppapaikkaan. Näin ollen esimerkiksi tietoverkon hitaus ei vaikuta agenttipohjaisen järjestelmän käytettävyyteen samalla tavalla kuten esimerkiksi WWW-sivujen kautta toteutettavan palvelun käytettävyyteen. Käyttäjän ei siis välttämättä tarvitse luoda yhteyttä mihinkään ennen kuin hän voi syöttää haluamiaan tietoja; perinteisessä elektronisessa kauppapaikassa käyttäjä ottaa aluksi yhteyden kauppapaikkaan ja alkaa sitten etsimään haluamiaan tuotteita. Perinteisessä, WWW-sivujen kautta käytettävässä kauppapaikassa yhteyden hitaus korostuu vielä erityisesti silloin, jos kauppapaikan WWW-sivuilla on käytetty runsaasti kuvia.

WWW-sivujen kautta toteutettavissa kauppapaikoissa palvelun käytettävyyteen vaikuttaa myös käyttäjän käytössä oleva selain. Johtuen useista eri selaimista (esimerkiksi Netscape, Internet Explorer, Lynx, Opera) ja niiden erilaisista ominaisuuksista, voi kauppapaikan ylläpitäjä joutua tinkimään kauppapaikan WWW-sivujen sisällöstä tai rajaamaan kauppapaikkansa toimivaan vain jollakin tietyllä selaimella.

Kummassakin tapauksessa ylläpitäjä joutuu tinkimään kauppapaikkansa käytettävyydestä. Agentin käyttöliittymä puolestaan voi muistuttaa enemmänkin jonkin ohjelman käyttöliittymää ja käyttäjä näkee käyttämästään käyttöjärjestelmästä tutut käyttöliittymäkomponentit. Näin ollen agentin käyttöliittymä ei ole esimerkiksi selainriippuvainen.

Kuvaton, pelkkiin käyttöliittymäkomponentteihin perustuva agentin käyttöliittymä ei välttämättä kuitenkaan ole yhtä tyylikäs kuin WWW-sivujen kautta toteutetulla kauppapaikalla vaan agentin käyttöliittymä voi olla asiallisen koruton. Käytettäessä WWW-sivujen kautta toteutettua kauppapaikkaa käyttäjä joutuu kuitenkin usein seuraamaan kauppapaikan käyttöliittymää koko ajan selatessaan kauppapaikassa tarjolla olevia tuotteita. Tällöin käyttöliittymän pitääkin olla mahdollisimman miellyttävä. Agentti puolestaan etsii tuotteet agentteihin pohjautuvassa kauppapaikassa käyttäjän puolesta, jolloin käyttäjä ei joudu seuraamaan agentin käyttöliittymää jatkuvasti. Seuraavan kerran käyttäjän täytyy kiinnittää huomiota agenttiin silloin, kun agentti palaa ja esittää etsinnän tulokset. Näin ollen agenttien käyttöliittymää seurataan huomattavasti lyhyempi aika kuin WWW-sivujen kautta toteutettujen kauppapaikkojen käyttöliittymää.

3.7. Agenteilla toteutettuja elektronisia kauppapaikkoja

Agentit ja elektroniset kauppapaikat ovat suhteellisen uusia ilmiöitä, joten agenteilla toteutettuja elektronisia kauppapaikkoja ei juuri ole olemassa. MIT:n media laboratoriossa on kehitetty joitakin vuosia sitten agentteihin perustuva

elektroninen kauppajärjestelmä Kasbah [MIT, 1997]. Kasbah:n toiminta on jo lopetettu, ja se on korvattu uudella järjestelmällä. Kasbah ja sen nykyinen seuraaja Market Maker [MIT, 1999] ovat ensimmäisiä agenteilla toteutettuja elektronisia kauppapaikkoja. Kasbah:sta kerrotaan tarkemmin kohdassa 3.7.1.

Kasbahin lisäksi yksi urauurtava agenteilla toteutettu elektroninen kauppapaikka on IBM:n kehittämä TabiCan¹ [IBM, 1997a], jonka välityksellä käyttäjä voi hankkia matkustamiseen liittyvää tietoa (matkalippujen hintoja ynnä muuta sellaista). TabiCan perustuu IBM:n e-Marketplace [IBM, 1997b] sovelluskehikseen (framework). e-Marketplace on toteutettu liikkuvilla agenteilla, tarkemmin sanottuna IBM:n kehittämällä agleteilla. e-Marketplace:sta kerrotaan tarkemmin kohdassa 3.7.2.

3.7.1. Kasbah

Kasbah [MIT, 1997] on tarkoitettu lähinnä yksittäisten tavaroiden myymiseen tai ostamiseen, joten se on suunnattu enemmän yksityisten kuluttajien väliseen kauppaan kuin jälleenmyyjille. Kasbahin seuraaja Market Maker puolestaan on tarkoitettu lähinnä opiskelijoille, jotka voivat myydä esimerkiksi kirjoja ja cd-levyjä toisilleen.

Chavez *et al* [1997] kertovat Kasbah-järjestelmän muodostuvan kahdesta osasta: käyttöliittymästä ja itse kauppapaikasta. Käyttöliittymä on *HTML*-pohjainen (HyperText Markup Language) ja se on toteutettu lähinnä Perl-skripteillä. Käyttöliittymän avulla käyttäjä pystyy hallinnoimaan omia agenttejaan. Lisäksi käyttöliittymän kautta voi lisätä käyttäjiä kauppapaikan käyttäjiksi.

Varsinaiset agentit luodaan itse kauppapaikassa, jossa agentit neuvottelevat muiden agenttien kanssa. Itse kauppapaikka, jossa agentit varsinaisesti käyvät neuvotteluita, on toteutettu Javalla. Näiden lisäksi on vielä tiettyihin erityistarkoituksiin olevia osia, kuten esimerkiksi tiedotusten lähettäminen hakulaitteisiin.

Chavez *et al* [1997] kuvaavat järjestelmän agentteja verrattain yksinkertaisiksi. Jotta agentit voisivat hoitaa niille annetun tehtävän, niiden täytyy tietää viisi asiaa: myytävä tavara, haluttu myyntihinta (tai ostohinta), alin hyväksyttävä hinta (tai korkein tarjottava hinta), aikaraja ja agentin strategia. Agentin strategialla voidaan määritellä miten agentin tavarasta haluama (tai tarjoama) hinta muuttuu halutusta alimpaan hyväksyttävään hintaan (tai korkeimpaan tarjottavaan hintaan). Agentin strategiaksi voidaan valita varovainen, kylmäpäinen tai ahne.

¹ Tabican on toteutettu vain japaniksi.

Chavez ja Maes [1996] kuvaavat agenttien strategioita hieman matemaattisemmin. Strategian mukaan agentin haluama hinta laskee (tai tarjoama hinta nousee) joko lineaarisesti, toisessa potenssissa tai kolmannessa potenssissa. Käyttäjä voi myös tarkistaa millaisia tarjouksia agentille on tehty ja tarvittaessa tehdä muutoksia agentin ominaisuuksiin (haluttuun hintaan, strategiaan ja niin edelleen).

3.7.2. e-Marketplace

e-Marketplace [IBM, 1997b] on liikkuvilla agenteilla toteutettu sovelluskehys elektronisille kauppapaikoille. e-Marketplacen pääasiallinen tehtävä on välittää käyttäjälle tietoa myytävänä olevista tuotteista. Nakamura ja Yamamoto [1997] määrittelevät e-Marketplacen seuraavasti: e-Marketplace on ympäristö, jossa tuottaja- ja kuluttaja-agentit voivat vaihtaa tietoa tuotteista. e-Marketplaceen perustuva järjestelmä voidaan yhdistää muihin e-Marketplaceen perustuviin järjestelmiin, jolloin saadaan käyttöön laajempi markkinointiverkosto.

e-Marketplacen arkkitehtuuri muodostuu seuraavista osista [Nakamura ja Yamamoto, 1997]:

- Agentti, joka suorittaa tehtävää käyttäjänsä puolesta. Järjestelmässä on kolmenlaisia agenteja: tuottaja-agentteja, kuluttaja-agentteja ja mainostaja-agentteja.
- *Ilmoitustaulu* (bulletin board), jolle agentit voivat laittaa ilmoituksia muille agenteille.
- *Ajastin* (agent timer), joka mahdollistaa agenttien toimimisen tietyn ajanjakson välein.
- *HTTP-rajapinta* (HyperText Transfer Protocol interface), joka tarjoaa käyttöliittymän WWW-selaimeen.
- *Agenttien valvoja* (agent scheduler), joka aktivoi ja deaktivoi agenteja tarpeen mukaan. Näin vältetään järjestelmän ylikuormittumista.
- *Viestimonitori* (message monitor), joka välittää kaikki agenttien, ilmoitustaulun, ajastimen ja HTTP-rajapinnan välillä liikkuvat viestit. Viestimonitorin tehtävä on ilmoittaa agenttien valvojalle milloin jonkin tietyn agentin kanssa halutaan kommunikoida.

Tuottaja-agenttien tehtävänä on myydä tai mainostaa tuotteita ja vastavasti kuluttaja-agentit toimivat tuottaja-agenttien asiakkaina. Mainostaja-agenttien tehtävänä on mainostaa toista e-Marketplaceen perustuvaa järjestelmää. Mainostaja-agentti voi ilmoittaa jonkin toisen e-Marketplacen ilmoitustaululla esimerkiksi jostakin oman järjestelmänsä tarjouksesta ja tehdä näin omaa

järjestelmäänsä tunnetuksi. Mainostaja-agentti voi mainostaa omaa järjestelmäänsä myös suoraan yhdelle agentille.

Mainostaja-agentteilla on tärkeä rooli useiden e-Marketplaceen perustuvien järjestelmien hyödyntämisessä. Ilman mainostaja-agentteja kuluttaja- tai tuottaja-agentit eivät osaisi siirtyä esimerkiksi juuri perustettuun e-Marketplaceen perustuvaan järjestelmään.

Nakamuran ja Yamamoton [1997] mukaan agenttien valvojan tehtävänä on aktivoida ja deaktivoida agentteja. Agentit aktivoidaan vain siksi aikaa, kun niiden kanssa kommunikoidaan; kommunikoinnin päätyttyä agentit jälleen deaktivoidaan. Aktivoimalla ja deaktivoidamalla agentteja säästetään järjestelmän resursseja ja vältetään järjestelmän ylikuormittumista.

Yksi järjestelmän tärkeimmistä osista on viestimonitori. Kaikki järjestelmän viestit kulkevat viestimonitorin kautta, joka välittää viestit eteenpäin ja suorittaa mahdolliset muut tarvittavat toimenpiteet. Viestimonitori esimerkiksi ilmoittaa aina agenttien valvojalle, milloin joku haluaa kommunikoida tietyn agentin kanssa. Agenttien valvoja puolestaan aktivoi kyseisen agentin, jotta kommunikointi olisi mahdollista. Kommunikoinnin päätyttyä viestimonitori ilmoittaa agenttien valvojalle kommunikoinnin päättyneen, jotta agenttien valvoja voisi deaktivoida agentin.

Koska viestimonitori välittää kaikki järjestelmässä liikkuvat viestit, voi se siis helposti päätellä milloin jonkin tietyn agentin kanssa kommunikoidaan. Viestimonitori on yhdessä agenttien valvojan kanssa vastuussa agenttien oikea-aikaisesta aktivoinnista ja deaktivoinnista.

Nakamuran ja Yamamoton [1997] mukaan e-Marketplacea voidaan käyttää joko WWW-selaimen tai erityisen asiakasohjelman kautta. Käytettäessä e-Marketplacea WWW-selaimen kautta agenttien lähettämät vastaukset näytetään joko HTML-muodossa tai lähetetään käyttäjälle sähköpostina. Käyttäjän ei tässä tapauksessa tarvitse asentaa mitään erityisohjelmistoja käyttääkseen e-Marketplacea, pelkkä WWW-selain riittää.

Halutessaan käyttäjät voivat asentaa koneeseensa erityisen e-Marketplace asiakasohjelmiston, jonka kautta käyttäjä voi luoda räätälöityjä agentteja järjestelmään. Asiakasohjelmiston avulla käyttäjä voi myös vastaanottaa järjestelmästä tulevia mainosagentteja.

3.8. Yhteenveto

Agentit voivat tukea käyttäjää monessa kaupankäynnin eri vaiheessa, esimerkiksi myyjän valinnassa. Kaupankäynnin vaiheista tarpeen tunnistaminen on

yksi vaikeimmista ohjelmallisesti toteuttavista, mutta yksinkertaisten tai rutinomaisten hankintojen kohdalla myös tätä vaihetta voitaisiin tukea.

Elektronisen kauppapaikan kehittäjälle ja ylläpitäjälle agentit voivat olla avuksi esimerkiksi käyttäjän tunnistamisessa.

Agenttien käyttäminen elektronisessa kauppapaikassa voi muuttaa tapaa, jolla elektronisia kauppapaikkoja käytetään: kauppapaikkaa ei enää käytetä WWW-selaimen kautta vaan agentit tuovat kauppapaikan myös käyttäjän koneelle. Hyvänä esimerkkinä tästä on e-Marketplace, jonka asiakasohjelmiston avulla käyttäjä voi jopa vastaanottaa kauppapaikasta tulevia mainosagentteja.

4. Liikkuvat agentit

Käsitteenä liikkuvat agentit on tunnettu jo useita vuosia [Harrison *et al*, 1995], mutta liikkuvat agentit eivät ole tulleet käytännössä erityisen suosituiksi. Vaikka staattisten agenttien määrä kasvaakin koko ajan, ei innostus ole levinnyt niin laajalti liikkuviin agentteihin. Syynä tähän voi olla esimerkiksi se, että liikkuvien agenttien toteutus tuntuu monimutkaiselta esimerkiksi staattisten agenttien toteuttamiseen verrattuna.

Suurin syy lienee kuitenkin se, että liikkuvat agentit asettavat staattisia agenteja enemmän vaatimuksia toimintaympäristölleen. Vaikka liikkuvien agenttien kehittämiseen tarkoitettuja kehitysympäristöjä kehitetään jatkuvasti, ei käyttökelpoisia kehitysympäristöjä vielä ole olemassa kovin montaa. Mikään näistä ei myöskään ole saavuttanut laajaa suosiota liikkuvien agenttien kehittäjien keskuudessa. Jos suurin osa liikkuvien agenttien kehittäjistä käyttäisi yhtä tiettyä kehitysympäristöä, olisi tällä järjestelmällä kehitetyillä agenteilla mahdollisuus liikkua useammilla palvelimilla ja järjestelmästä saataisiin enemmän hyötyä.

Tällä hetkellä näin ei kuitenkaan ole, vaan kaikilla eri kehitysympäristöillä on verrattain pieni käyttäjäkunta. Eri kehitysympäristöjen yhteistoiminnallisuutta on kuitenkin pyritty parantamaan esimerkiksi kehittämällä eri agenttijärjestelmien välisiä rajapintoja. Tätä aihetta käsitellään tarkemmin kohdassa 4.6.1.

Seuraavassa käydään läpi liikkuviin agentteihin liittyviä määritelmiä ja liikkuvien agenttien tärkeimpiä ominaisuuksia. Myös agenttien liikkuvuudella saavutettavia etuja ja siitä koituvia haittoja käsitellään. Tämän jälkeen käydään läpi muutamia olemassa olevia liikkuvien agenttien kehittämiseen tarkoitettuja kehitysympäristöjä ja vertaillaan näitä kehitysympäristöjä keskenään.

4.1. Itsenäisen liikkuvan agentin määritelmä

Venners [1997] on määritellyt agentteihin liittyviä termejä varsin kansantajuisesti. Vennersin mukaan liikkuvalla agentilla tarkoitetaan sellaista ohjelmaa, joka voi keskeyttää suorituksensa, siirtää itsensä verkon välityksellä toiselle koneelle ja jatkaa suoritustaan siitä, mihin se aikaisemmin jäi. Itsenäisellä agentilla tarkoitetaan puolestaan sellaista agenttia, joka päättää itse mille palvelimelle se siirtyy ja mitä se tekee. Itsenäinen agentti saa pyyntöjä ulkoisista lähteistä,

esimerkiksi muilta agenteilta, mutta kukin agentti päättää, vastaako se näihin pyyntöihin vai ei.

Schubert *et al* [1998] määrittelevät itsenäisen agentin ja sen ominaisuudet hieman tarkemmin. Heidän mukaansa itsenäisellä agentilla on tarpeeksi älyä, jotta se voisi tehdä omia johtopäätöksiä ja hoitaa tehtäväänsä itsenäisesti. Itsenäisen agentin tulisi kommunikoida käyttäjänsä kanssa vain tärkeissä päätöksissä. Jotta itsenäiset agentit voisivat suoriutua tehtävistään ja jotta niistä olisi myös hyötyä käyttäjilleen, on agenttien omattava jonkinlaista asiantuntemusta. Agentin tulee myös pystyä kommunikoimaan ympäristönsä kanssa, jotta se voisi hankkia päätöksissään tarvitsemaansa tietoa.

Schubert kollegoineen määrittelee itsenäiselle agentille varsin tärkeitä ominaisuuksia, kuten esimerkiksi älyn omaaminen. Agentin ei kuitenkaan tarvitse omata varsinaista (teko)älyä ollakseen itsenäinen, vaan riittää, että agentti osaa reagoida tietyn tyyppisiin tilanteisiin. Schubert mainitsee myös, että itsenäinen agentti kommunikoi käyttäjänsä kanssa vain tärkeissä päätöksissä. Kommunikointi käyttäjän kanssa on kuitenkin varsin järjestelmäkohtaista. Elektronisissa kauppasovelluksissa tällainen tärkeä päätös voi olla esimerkiksi ostopäätös. Viime kädessä ostopäätöksen pitää olla käyttäjän päätös, eikä agentin tulisi tehdä sitä.

Schubert *et al* [1998] kertovat agenttien yleisimmäksi käyttötarkoitukseksi käyttäjän auttamisen aikaa kuluttavissa tiedonhakutehtävissä. Schubert kollegoineen ennustaa, että tulevaisuudessa agentit tulevat hoitamaan kokonaisuudessaan sekä rutiininomaisia että tiedonvaihtoon perustuvia tehtäviä: agenteista tulee yksi tärkeimmistä tekniikoista tulevaisuuden digitaalisen kaupankäynnin ratkaisuisissa.

Agentteja voidaan tarkastella edellä mainitun, verrattain käyttäjakeskeisen lähestymistavan lisäksi myös järjestelmän näkökulmasta. Lange ja Oshima [1998] esittävät järjestelmän näkökulman agenttiin. Järjestelmän näkökulmasta agentti on olio, joka on sijoittunut johonkin suoritussympäristöön. Jotta olio olisi agentti, tulisi tämän olla reaktiivinen, autonominen ja päämäärätietoinen. Näiden ominaisuuksien lisäksi olio voi olla myös liikkuva, oppiva ja kommunikoi-va.

4.2. Agenttien tärkeitä ominaisuuksia

Agentin ominaisuudet riippuvat suuresti agentin käyttötarkoituksesta: käyttöliittymäagentin ominaisuudet eroavat suuresti esimerkiksi tiedonhakuagentin ominaisuuksista. Edellä mainittiin muutamia ominaisuuksia, joita ilman oliota ei voida laskea agentiksi. Muun muassa nämä ominaisuudet ovat yhteisiä

kaikentyyppisille agenteille. Agentin tulisikin omata ainakin jollakin tasolla eräitä perusominaisuuksia, jotta siitä voisi olla hyötyä käyttäjälleen. Schubert *et al* [1998] luettelevat tällaisiksi ominaisuuksiksi seuraavat:

- *Reaktiivisuus*, jolla tarkoitetaan agentin kykyä reagoida ympäristössään tapahtuviin muutoksiin. Reagoimisen edellytyksenä on tietenkin, että agentti myös jollakin tavalla seuraa ympäristönsä tapahtumia.
- *Päämäärätietoisuus*, jolla tarkoitetaan, että agentilla on tietty tehtävä, jota se suorittaa. Agentin päämäärätietoisuus vaikuttaa myös siihen, miten agentin tulisi reagoida ympäristössään tapahtuviin muutoksiin.
- *Kyky oppia ja sopeutua* on tärkeää, jotta agentti kykenisi toimimaan käyttäjänsä tavoin. Agentin tulisi siis kerätä riittävästi tietoa, jotta se voisi oppia riittävästi käyttäjänsä tavoista toimiakseen käyttäjänsä edustajana. Agentin tulisi myös kommunikoida muiden agenttien kanssa ja oppia kokemistaan asioista. Käytännössä kuitenkin todellisen oppivan agentin toteuttaminen on enemmänkin tekoälytekniikkaa kuin rutiininomaisien tehtävien hoitamiseen suunniteltujen agenttien toteuttamista. Agentin oppiminen on kuitenkin yksi agentin tärkeimmistä ominaisuuksista, mutta samalla se on myös vaikeimmin toteutettavissa.
- *Autonomisuus*, jolla tarkoitetaan sitä, että agentti suorittaa tehtävänsä itsenäisesti, ilman käyttäjän suoraa ohjausta. Agentin tulisi siis omata riittävästi tietoa, jotta se voi toimia itsenäisesti. Voidakseen toimia itsenäisesti agentin täytyy olla myös tietoinen sen päämäärästä.
- *Liikkuvuus*, jolla tarkoitetaan agentin kykyä siirtyä elektronisessa tietoverkossa. Liikkuva agentti voi missä tahansa suorituksensa vaiheessa siirtyä tietoverkossa toiselle palvelimelle. Agentin mukana siirtyä myös sen suorituksen tila, jotta agentti voi siirtymisen jälkeen jatkaa suorituksesta.
- *Kommunikointi ja yhteistyö* ovat tärkeimpiä ominaisuuksia etenkin moniagenttijärjestelmissä. Agentin kyky kommunikoida luo pohjan älykäsagenttien toiminnalle, ilman kommunikointikykyä agentti ei pysty hankkimaan tietoa ympäristöstään. Kommunikointi on myös edellytys agenttien yhteistyölle. Kommunikointia voidaan tarvita agenttityypistä riippuen paitsi agenttien välillä, niin mahdollisesti myös agentin ja käyttäjän välillä.

Koska agenttien käyttötarkoitus vaihtelee suuresti, ei yksittäisen agentin tarvitse omata kaikkia näitä ominaisuuksia. Laajemmassa agenttijärjestelmässä tehtäviä voidaan jakaa agenttien kesken siten, että yksittäisen agentin täytyy

omata ominaisuuksista vain muutama: jokin agentti voi keskittyä vain ympäristön tarkkailuun, kun taas toinen agentti voi kerätä järjestelmälle tietämyskantaa. Laajassa järjestelmässä kaikkien agenttien tulisi kuitenkin omata kyky kommunikoida ja tehdä yhteistyötä toisten agenttien kanssa.

4.3. Liikkuvien agenttien vertailu paikallaan pysyviin agentteihin

Kaikkien agenttien toimintaperiaatteen voisi yleistää siten, että niiden keskeisenä tarkoituksena on suorittaa tiettyjä käyttäjän määrittelemiä toimenpiteitä käyttäjänsä puolesta. Vaikka tämä toimintaperiaate onkin sama sekä liikkuville että staattisille agenteille, eroavat ne huomattavasti tavassa, jolla ne tätä toimintaperiaatetta toteuttavat.

Staattisten agenttien toimintaympäristö on tietokone, jossa se prosessoi tietoa. Agentin tyypistä ja tarkoituksesta riippuen staattinen agentti noutaa tarvitsemansa tiedon joko koneesta, johon se on sijoittunut, tai tietoverkosta, jos kone on sellaiseen liitetty. Staattisten agenttien suurin hyöty onkin tiedon prosessoiminen ja tehtävien hoitaminen käyttäjän puolesta.

Liikkuvien agenttien toimintaympäristö on tietoverkko. Staattisesta agentista poiketen liikkuva agentti ei nouda tarvitsemaansa tietoa tietoverkosta vaan siirtyy tiedon luokse, prosessoi tiedon ja siirtyy takaisin. Liikkuvien agenttien suurin hyöty onkin juuri niiden liikkuvuudessa, muuten ne pystyvät suorittamaan samoja tehtäviä kuin staattiset agentitkin.

Agenttien liikkuvuuden edellytyksenä on kuitenkin se, että agentin kohdepalvelimessa on agentille sopiva alusta. Tämä on välttämätöntä, jotta agentti kykenisi hankkimaan tietoa palvelimelta tai palvelimen välityksellä. Liikkuvat agentit ovat siis hyvin pitkälle alustariippuvaisia, eivätkä siis voi siirtyä mille tahansa palvelimelle.

Staattiset agentit ovat liikkuvia agenteja yleisempiä myös siitä syystä, että staattiset agentit eivät välttämättä tarvitse liittymää tietoverkkoon. Staattisten agenttien tietoverkkoliittymän tarve riippuu agentin käyttötarkoituksesta, mutta liikkuville agenteille tietoverkko on välttämätön; muuten agenttien liikkuvuudesta ei ole mitään hyötyä. Koska etukäteen ei välttämättä tiedetä milloin tietoverkkoon siirtynyt liikkuva agentti saa tehtävänsä suoritettua, ja olisi valmis palaamaan tietokoneelle, josta se lähetettiin, olisi tietoverkkoliittymän lisäksi hyvä olla kiinteä. Koska läheskään kaikissa tietokoneissa ei vielä ole kiinteää liittymää tietoverkkoon, on myös liikkuvista agenteista hyötyviä käyttäjiä pieni osa kaikista tietokoneen käyttäjistä.

4.4. Agenttien liikkuvuudesta saatavia hyötyjä

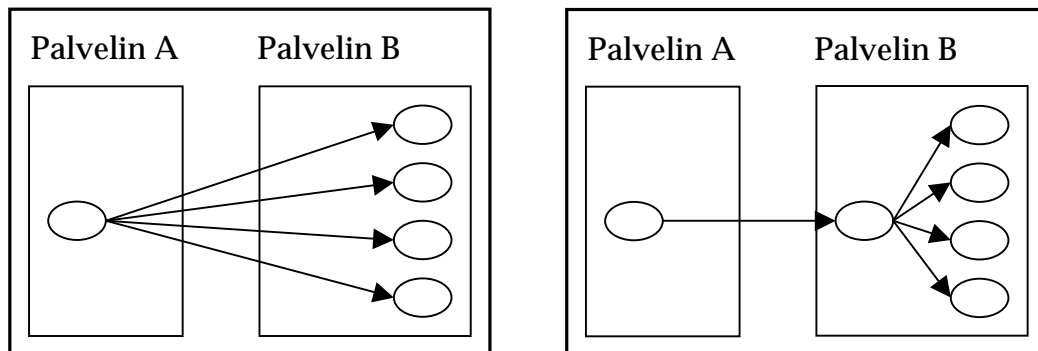
Koska liikkuvilla agenteilla on mahdollisuus siirtyä toiselle palvelimelle jatkamaan suoritustaan, voidaan tästä hyötyä tietynlaisissa sovelluksissa. Useimmat tehtävät, joissa agenttien liikkuvuudesta voidaan hyötyä, ovat luonteeltaan sellaisia, että ne vaatisivat staattiselta agentilta yhteydenottoa useisiin muihin palvelimiin ja hakemaan niistä mahdollisesti suurenkin määrän tietoa.

4.4.1. Verkon kuorman väheneminen

Liikkuvilla agenteilla voidaankin suorittaa tietoverkossa tietynlaisia tehtäviä huomattavasti tehokkaammin kuin staattisilla agenteilla. Liikkuvat agentit voivat siirtyä toiselle palvelimelle suorittamaan omaa tehtäväänsä, toisin kuin staattiset agentit, jotka toimivat omalta palvelimeltaan.

Liikkuvat agentit voivat siis esimerkiksi siirtyä prosessoimaan vieraalta palvelimelta löytyvää tietoa itse palvelimelle ja siirtää kotipalvelimelleen vain prosessoinnin tulokset. Vastaavassa tilanteessa staattinen agentti siirtäisi prosessoitavan tiedon ensin kotipalvelimelleen ja prosessoisi tiedon siellä. Siirtymisen tiedon tarjoavalle palvelimelle on erityisen hyödyllistä, jos kyseessä ovat suuret tietomäärät. Liikkuvien agenttien avulla verkossa turhaan liikkuvan tiedon määrää voitaisiin pienentää huomattavasti.

Myös hajautetussa moniagenttijärjestelmässä verkon kuormaa pystytään vähentämään ottamalla käyttöön liikkuvat agentit. Sen sijaan, että kotipalvelimessa oleva staattinen agentti ottaisi yhteyden useaan toisessa palvelimessa olevaan agenttiin, voisi liikkuva agentti siirtyä toiseen palvelimeen ja muodostaa yhteyden palvelimen sisällä vähentäen verkon kuormaa.



Kuva 1. Käytettäessä liikkuvia agentteja staattisten agenttien sijaan voidaan vähentää verkon kautta otettavien yhteydenottojen määrää.

4.4.2. Riippumattomuus protokollista

Langen ja Oshiman [1998] mukaan liikkuvilla agenteilla voidaan myös saavuttaa riippumattomuus protokollista. Siirrettäessä tietoa hajautetussa järjestelmässä tarvitaan siihen useimmiten jonkinlainen protokolla. Tiedon lähettävä osapuoli käsittelee tiedon tietyllä tavalla ja vastaavasti vastaanottaja tietää miten tieto pitää purkaa. Jos käytettävään protokollaan tulee muutoksia – esimerkiksi turvallisuuden lisäämiseksi – pitää hajautetussa järjestelmässä muuttaa jokaista tiedon lähettävää tai vastaanottavaa ohjelmaa.

Protokollien sijaan voidaan käyttää liikkuvia agentteja. Agentti voi kuljettaa mukanaan esimerkiksi salakirjoitetun tiedon, purkaa sen määränpäässään ja "luovuttaa" selväkielisen tiedon määränpäähensä. Mahdolliset muutokset tulee tällöin tehdä vain tiedon kuljettaviin agentteihin.

4.4.3. Laajemmat oikeudet

Liikkuva agentti voi myös hyötyä siirtyessään tiedon tarjoavalle palvelimelle saamalla laajemmat oikeudet kuin vieraalta palvelimelta toimiva staattinen agentti. Tämä voi olla erityisen hyödyllistä etenkin palvelimen tilaa monitoroivissa agenttisovelluksissa, joissa agentin kannalta tärkeät tiedot eivät ole kaikkien saatavilla.

Liikkuvan agentin siirtyessä toiselle palvelimelle sille voidaan esimerkiksi myöntää oikeus lukea palvelimen levyltä jokin tietty tiedosto, mahdollisesti jopa kaikki tiedostot. Vastaavasti voidaan myöntää myös kirjoitusoikeuksia. Tällaisia oikeuksia ei välttämättä myönnettäisi toiselta palvelimelta yhteydenottavalle staattiselle agentille.

4.4.4. Palvelujen saavutettavuus

Lange ja Oshima [1998] esittävät tilanteita, joissa liikkuvilla agenteilla pystytään parantamaan palvelun saavutettavuutta. Jos esimerkiksi jokin agentille tarjottu palvelu on huonon verkkoyhteyden takana, ei staattinen agentti välttämättä pysty hoitamaan tehtävänsä, etenkin jos sen pitäisi noutaa palvelusta tietoa usein. Liikkuvan agentin toimintaa huono verkkoyhteys ei haittaa näin paljon, koska kun liikkuva agentti on kerran päässyt siirtymään palvelun tarjoavalle koneelle, pystyy se hoitamaan tehtävänsä koneella autonomisesti huonosta verkkoyhteydestä huolimatta. Kun tieto on prosessoitu, siirtyy agentti takaisin omalle kotikoneelleen, kun verkkoyhteys on jälleen kunnossa.

Langen ja Oshiman [1998] mukaan agenteilla pystytään myös poistamaan tietoverkon mahdollisista viiveistä aiheutuvat haitat. Ohjattaessa esimerkiksi tehtaan robotteja tietoverkon kautta on usein tärkeää, että kontrolliviestit saapuvat robotille viiveettä. Suuressa verkossa on usein kuitenkin viiveitä ja nämä viiveet voivat olla prosessin kannalta kohtalokkaita. Tällaisessa tilanteessa liikkuva agentti voidaan siirtää ohjaamaan robottia paikallisesti, jolloin verkon viiveet eivät hidasta kontrolliviestejä.

Käytettäessä huonon tai hitaan verkkoyhteyden päässä olevaa palvelua, voitaisiin palvelun saavutettavuutta parantaa siirtymällä käyttämään palvelua liikkuvan agentin kautta. Näin käyttäjän ei tarvitsisi kärsiä hitaasta verkkoyhteydestä, vaan käyttäjä käyttäisi omalla koneellaan olevaa agenttia, joka puolestaan huolehtisi kommunikoinnista varsinaisen palvelun tarjoavan palvelimen kanssa. Agentilla olisi käyttöliittymä, joka toimisi siis samalla myös koko palvelun käyttöliittymänä ja vain olennainen tieto - ei kuvia eikä turhaa tekstiä - haettaisiin verkossa olevalta palvelimelta. Aloittaessaan palvelun käytön käyttäjä saisi heti näkyviinsä valmiin käyttöliittymän, eikä hänen tarvitsisi odottaa itselleen mahdollisesti epäolennaisen tiedon, esimerkiksi joidenkin kuvien, latautumista.

4.4.5. Reagoiminen virhetilanteisiin

Lange ja Oshima [1998] mainitsevat liikkuvien agenttien eduiksi myös niiden mahdollisuudet reagoida virhetilanteisiin. Liikkuvien agenttien on mahdollista reagoida erilaisiin virhetilanteisiin monipuolisemmin kuin staattisten agenttien tai muiden ohjelmien. Esimerkiksi sammutettaessa konetta voivat agentit saada tästä ilmoituksen ja siirtyä jatkamaan suoritustaan toiselle koneelle. Staattinen agentti voi vastaavassa tilanteessa kirjoittaa tilansa levyille ja jatkaa suoritustaan vasta, kun palvelin on käynnistetty uudelleen.

Jotta liikkuva agentti voisi siirtyä pois esimerkiksi sammutettavalta palvelimelta, edellyttää se kuitenkin, että agentilla on tiedossaan jonkin toisen palvelimen osoite. Jos agentilla ei ole tiedossaan minkään muun palvelimen osoitetta tai siirtyminen ei jostakin syystä onnistu, täytyy agentin turvautua samanlaisiin keinoihin esimerkiksi staattisen agentin kanssa ja kirjoittaa tilansa levyille.

4.4.6. Hajautettujen järjestelmien ohjaus

Hajautettujen järjestelmien ohjaus voidaan toteuttaa helpommin liikkuvilla agenteilla kuin esimerkiksi ohjausviesteillä. Järjestelmää ohjaavalta koneelta ei tarvitse koko ajan lähettää ohjausviestejä järjestelmän muille koneille, vaan ohjaava kone voi siirtää muille koneille agentin, joka ohjaa muita järjestelmän koneita. Näin voidaan säästää sekä verkon että järjestelmää ohjaavan koneen resursseja.

Liikkuvia agentteja voidaan hyödyntää tällaisessa tilanteessa etenkin silloin, jos ohjausviestit noudattavat jotakin tiettyä kaavaa, liikkuva agentti voisi ohjata esimerkiksi liukuhihnalla työskentelevää robottia. Tällöin agentti voidaan ohjelmoida ohjaamaan konetta kaavan mukaan.

4.5. Agenttien liikkuvuuden ongelmia

Paitsi hyötyä, agenttien liikkuvuudesta voi olla myös haittaa. Tuomalla järjestelmään liikkuvia agentteja järjestelmä voi monimutkaistua sekä vaatia enemmän ylläpitoa. Liikkuvat agentit vaativat usein toimiakseen tietynlaisen alustan, joka voi myös lisätä ylläpidon tarvetta. Haittaa voivat aiheuttaa myös huonosti ohjelmoidut agentit. Seuraavassa käydään tarkemmin läpi agenttien liikkuvuuden mukanaan tuomia mahdollisia ongelmia ja haittoja.

4.5.1. Yhtenäisen järjestelmän tarve

Yksi keskeinen agenttien liikkuvuudesta aiheutuva ongelma on yhtenäisen järjestelmän tarve. Jos käyttäjä ottaa käyttöönsä esimerkiksi jonkinlaisen käyttäjän toimia seuraavan staattisen agentin, joka käynnistyy ja toimii hänen koneellaan, tietää käyttäjä, että agentilla on riittävä ja oikeanlainen toimintaympäristö. Käyttäjälle riittää, että staattinen agentti toimii hänen koneellaan eikä sen tarvitse toimia muualla. Riittää siis, että juuri käyttäjän koneessa on oikeanlainen järjestelmä.

Jotta liikkuvien agenttien liikkuvuudesta olisi hyötyä, täytyy kuitenkin käyttäjän koneen lisäksi useassa muussakin koneessa olla käytössä samanlainen järjestelmä. Jos käyttäjä on ainut, jolla on liikkuvien agenttien vaatima järjestelmä käytössään, ei liikkuva agentti pysty siirtymään millekään muulle palvelimelle eikä näin ollen toimimaan.

Koska vielä ei ole mitään laajalle yleistynyttä liikkuvien agenttien agenttialustaa, agentit eivät voi siirtyä kuin pienelle osalle palvelimista. Jotta

liikkuvista agenteista voitaisiin saada suurempi hyöty ja jotta liikkuvat agentit yleistyisivät, pitäisikin saada laajalti käyttöön jokin helposti omaksuttava agenttialusta.

4.5.2. Järjestelmän vaatima ylläpito

Palvelimelle asennettava agenttialusta vaatii tietysti ylläpidolta huomiota. Jos palvelimen agenttialustalle siirtyy suuri määrä agentteja, lisää se myös palvelimen kuormitusta. Jos palvelin on ylikuormitettu, ei se välttämättä pysty hoitamaan kaikkia tehtäviään sille tarkoitettulla tavalla ja näin ollen antaa huonon kuvan palvelimesta käyttäjälle. Ylläpidon täytyy siis jotenkin kontrolloida myös järjestelmän käyttämiä resursseja.

Resurssien valvonta on tärkeää paitsi käytettävyy- ja luotettavuussyistä myös turvallisuuden kannalta, sillä toisinaan hakkerit yrittävät murtoyrityksen yhteydessä kuormittaa palvelinta mahdollisimman paljon. Ilman resurssien valvontaa palvelimen kuormittaminen agenttijärjestelmän avulla olisi helppoa: pitäisi vain lähettää palvelimelle suuri määrä agentteja.

4.5.3. Järjestelmien monimutkaistuminen

Käytettäessä liikkuvia agentteja toteutettavan järjestelmän osana järjestelmästä voi tulla monimutkainen. Sen jälkeen, kun järjestelmän lähettämä agentti on siirtynyt toiselle palvelimelle suorittamaan tehtäväänsä, ei järjestelmä välttämättä saa tietoa agentista ennen kuin agentti saa tehtävänsä suoritettua ja palaa kotipalvelimelleen. Agentti voi siirtyä tehtävänsä aikana mahdollisesti useita kertoja, riippuen suuresti tehtävän luonteesta. Myöskään agentin paluustajasta ei välttämättä ole tietoa, etenkin jos tehtävä vaatii agentilta useita siirtymiä palvelimelta toiselle.

Jos järjestelmän sammuttaminen on mahdollista milloin tahansa, tulisi järjestelmän tällöin myös kutsua lähettämänsä agentit takaisin. Koska agentti on voinut siirtyä tehtävänsä aikana jo useita kertoja, ei agentin sijainnista ole välttämättä tarkkaa tietoa. Tällaisten tilanteiden varalta agentin tulisi ilmoittaa järjestelmälle mihin se on siirtynyt. Järjestelmän tulee siis pitää kirjaa kaikkien lähettämiensä agenttien siirtymisistä ja nykyisistä sijainneista.

4.5.4. Agenttien villiintyminen

Jos agenttia ei suunnitella ja toteuteta erityisen huolellisesti, voi agentti villiintyä eli lähteä kiertämään verkkoa hallitsemattomasti. Jos agentti esimerkiksi ei pysty suorittamaan tehtäväänsä, eikä sille ole asetettu määräaikaa tehtävän suorittamiseen, voi agentti jäädä kiertämään verkkoa yrittäen ratkaista tehtävänsä. Käyttäjä on voinut esimerkiksi epähuomiossa antaa agentille mahdolloman tehtävän ratkaistavaksi, eikä agentin ohjelmoija ole ottanut tällaista tilannetta huomioon. Tällöin agentti aiheuttaa ylimääräistä kuormaa sekä verkolle että palvelimille. Agentteja ohjelmoitaessa olisikin noudatettava erityistä huolellisuutta ja varmistettava, että agentti ei voi joutua sellaiseen tilanteeseen, että se vain siirtyy eri palvelimien välillä saamatta mitään aikaan.

4.6. Kehitysympäristöjä liikkuvien agenttien toteuttamiseen

Suurin valmiista kehitysympäristöstä saatava hyöty on se, että ohjelmoijan ei tarvitse itse keskittyä esimerkiksi agentin liikkuvuuden toteuttamiseen, vaan hän voi keskittyä sisällön tuottamiseen. Liikkuvuuden lisäksi kehitysympäristössä on pääsääntöisesti jo luotu toimivat ratkaisut esimerkiksi viestinvälitykselle, joten agenttien ohjelmoijan ei tarvitse kiinnittää huomiota myöskään tähän ehkä ongelmalliseenkin liikkuvien agenttien osa-alueisiin. Lisäksi joissakin järjestelmissä on voitu toteuttaa joitakin järjestelmäspesifisiä ratkaisuja helpottamaan agenttien kehittäjän työtä.

Liikkuvien agenttien luomiseen on kehitetty useita kehitysympäristöjä. Kukin kehitysympäristö on toteutettu eri tavalla, joten järjestelmät eivät ole yhteensopivia. Järjestelmien yhteistoiminnallisuutta on kuitenkin pyritty parantamaan järjestelmien välisillä rajapinnoilla. Seuraavassa käsitellään aihetta lisää.

Seuraavassa esitellään lyhyesti myös joitakin liikkuvien agenttien kehitysympäristöjä ja näiden lisäksi luvussa viisi tarkastellaan lähemmin IBM:n kehittämää Aglets-järjestelmää. Seuraavassa esiteltävät agenttien kehitysympäristöt ovat kehitysympäristöjä Java-kieleen pohjautuvien liikkuvien agenttien kehittämiseen. Näiden lisäksi esitellään vielä staattisten agenttien kehittämiseen tarkoitettu JATLite-järjestelmä sen edistyksellisen viestinvälitysjärjestelmän vuoksi.

4.6.1. MASIF

Yksi liikkuvien agenttien yleistymistä hidastava seikka on se, että eri agenttijärjestelmät eivät ole yhteensopivia keskenään. Agenttijärjestelmien yhteistoiminnallisuuden parantamiseksi on kehitetty agenttijärjestelmien välistä standardoitua rajapintaa *MASIF*:ia (Mobile Agent System Interoperability Facility). *MASIF*a kehittää *Object Management Group* (OMG) ja siihen viitataan usein nimellä *OMG MASIF*.

Lange ja Oshima [1998] kirjoittavat myös *MASIF*:ista. *MASIF* siis määrittelee agenttijärjestelmien välisen rajapinnan, jotta agenttijärjestelmien välinen yhteistoiminta olisi mahdollista. *MASIF* ei kuitenkaan mahdollista eri ohjelmointikielillä toteutettujen agenttijärjestelmien välistä yhteistoiminnallisuutta, vaan rajoittuu ainoastaan samalla ohjelmointikielellä toteutettujen agenttijärjestelmien väliseen yhteistoiminnallisuuteen.

MASIF:in tarkoituksena ei ole standardoida tapaa, jolla yksittäinen agenttijärjestelmä käsittelee järjestelmän agentteja, vaan yhtenäistää agenttien käsittelytapaa järjestelmätasolla. *MASIF*issa käsiteltävät neljä eri kohtaa ovat [Lange ja Oshima, 1998]:

- *Agenttien hallinta*, jonka tarkoituksena on yhtenäistää tapa, jolla järjestelmän ylläpitäjät voivat hallinnoida eri järjestelmien agentteja. *MASIF*-yhteensopivassa järjestelmässä tulisi olla mahdollista muun muassa etsiä agentteja, luoda agentteja niiden luokkanimen perusteella ja niin edelleen.
- *Agenttien siirtyminen*, jonka tavoitteena on mahdollistaa agenttien siirtyminen myös sellaisiin palvelimiin, joissa on jonkin muun *MASIF*-yhteensopivan agenttijärjestelmän palvelinohjelmisto.
- *Agenttien ja agenttijärjestelmien nimeäminen*, jonka tavoitteena on yhtenäistää agenttien ja agenttijärjestelmien nimeämistä niin, että agentit ja agenttijärjestelmät voisivat helpommin tunnistaa toisensa.
- *Agenttijärjestelmien tyyppi- ja sijaintitiedot*, jonka tarkoituksena on yhtenäistää tyyppi- ja sijaintitietoja. Jotta agentti voisi siirtyä palvelimelle, pitää sen tietää tukeeko palvelimella oleva palvelinohjelmisto tämän tyyppisiä agentteja. Yhtenäistämällä eri agenttijärjestelmien tyyppitietoja tämä tieto on helpommin agenttien saatavilla. Sijaintitietojen yhtenäistäminen on tarpeen, jotta järjestelmät voisivat paikallistaa toisensa.

4.6.2. Java-To-Go

Li ja Messerschmitt [1996] kuvaavat Java-To-Go-järjestelmää kokeelliseksi järjestelmäksi, jonka tavoitteena on auttaa ohjelmoijia kokeilemaan ja kehittämään liikkuvia agentteja ja hajautettuja, agentteihin pohjautuvia sovelluksia. Java-To-Go on nimensä mukaisesti Java-kieleen perustuva järjestelmä. Java-To-Go tarjoaa ympäristön, jossa Java-kielellä toteutettuja liikkuvia agentteja voidaan siirtää eri palvelimien välillä.

Java-To-Go perustuu kahdenlaisiin komponentteihin: liikkuviin agentteihin ja staattisiin palvelinohjelmistoihin, joita järjestelmässä kutsutaan nimellä *hall server*. Jokaisessa palvelimessa voi olla yksi tai useampi *yhteisö* (community), jossa puolestaan on yksi tai useampi palvelinohjelmisto. Näin muodostuu verkko palvelinohjelmistoja, joiden välillä järjestelmän agentit voivat liikkua. Jokainen agentti koostuu kolmesta osasta: luokkamäärittelystä, agentin tilan määrittelevistä muuttujista ja listasta, johon on tallennettu palvelinohjelmistojen sijainteja. Palvelinohjelmistojen sijainteja sisältävä lista muodostaa reitin, jota pitkin agentin tulisi kulkea määränpäähänsä. Agentti voi hankkia tietoa palvelimesta palvelimella olevan palvelinohjelmiston kautta. Agentin tila ja palvelinohjelmistojen sijainteja sisältävä lista ovat dynaamisia eli agentti voi muuttaa niitä matkansa aikana; agentin luokkamäärittely on staattinen. Päästyään määränpäähänsä agentti "julkaisee" reitin varrella keräämänsä tiedot ("tulokset").

Java-To-Go-järjestelmän agentit eivät voi keskeyttää suoritustaan milloin tahansa jatkaakseen sitä toisella palvelimella. Siirtyessään palvelimelta toiselle agentin tila voidaan kuitenkin säilyttää erityisten tilamuuttujien avulla. Tilamuuttujien tallennusta varten agentilla on olemassa ns. boot- ja wrapup-koodit. Lähtiessään palvelimelta agentti tallentaa suorituksensa tulokset tilamuuttujiin (wrapup). Saapuessaan palvelimelle yritetään agentin tila palauttaa sen tilamuuttujien perusteella (boot).

Palvelinohjelmisto tarjoaa agenteille vain yleisimmin tarvittavat palvelun LocalAccess-rajapinnan kautta. Palvelinohjelmiston toiminnallisuutta voidaan laajentaa ohjelmoimalla palvelinohjelmistoon lisää agenteille tarjottavia, tiettyyn tarkoitukseen soveltuvia rajapintoja.

4.6.3. Mole

Pfisterer [1998] kuvaa Stuttgartin yliopistossa kehitettyä Mole-järjestelmää Java-kieleen pohjautuvaksi liikkuvien agenttien järjestelmäksi. Järjestelmä

koostuu kahdenlaisista agenteista: *järjestelmäagenteista* (system agents) ja *käyttjäagenteista* (user agents).

Järjestelmäagenteilla on muita agentteja suuremmat oikeudet, niillä on esimerkiksi pääsy palvelimen tiedostojärjestelmään. Järjestelmäagentit toimivat usein rajapintaolioina agenttijärjestelmän ulkopuolella oleviin resursseihin, esimerkiksi tietokantoihin. Järjestelmäagentit siis tarjoavat muille agenteille hallitsemansa resurssin palveluita. Järjestelmäagentit luodaan järjestelmän käynnistyessä, ja ne ovat olemassa niin kauan kunnes järjestelmä sammutetaan. Järjestelmäagentit ovat yleensä staattisia eli ne eivät voi siirtyä toiselle palvelimelle. Järjestelmäagentin siirtyminen toiselle palvelimelle ei usein olisikaan kovin järkevää, sillä järjestelmäagentti voi olla toteutettu toimimaan vain siinä ympäristössä, jossa se on kehitetty. Esimerkiksi tietokantaa hallinnoivan järjestelmäagentin täytyy tietää millaista tietokantaa se hallinnoi, missä sen hallinnoima tietokanta sijaitsee ja millaista tietoa järjestelmäagentin tulisi tarjota muille agenteille. Tällaisesta järjestelmäagentista ei ole hyötyä missään muussa järjestelmässä, ellei järjestelmässä ole samanlaista tietokantaa samassa paikassa.

Käyttjäagentit puolestaan voivat siirtyä palvelimien välillä, mutta niillä on pienemmät oikeudet kuin järjestelmäagenteilla. Pienemmistä oikeuksistaan huolimatta käyttjäagentit voivat kommunikoida toisten agenttien ja käyttäjän kanssa sekä luoda toisia agentteja. Jokaisella järjestelmän agentilla on yksilöllinen tunniste, jota käytetään pääasiassa agentin kanssa kommunikoidessa.

Palvelimella sijaitseva agenttien ympäristö on Mole-järjestelmässä nimeltään *sijainti* (location). Yhdellä palvelimella voi olla monta sijaintia, joita kaikkia palvelee yksi Mole-järjestelmän *moottori* (engine).

4.6.4. Concordia

Concordia [Mitsubishi, 1998] esitellään Java-kielellä toteutetuksi agenttikehykseksi, jolla voidaan kehittää liikkuvia agentteja. Concordia-järjestelmän agentit liikkuvat palvelimilla olevien palvelinohjelmistojen, joita Concordia-järjestelmässä kutsutaan nimellä Concordia Server, välillä. Palvelimella sijaitsevat palvelinohjelmistot huolehtivat agenttien elinkaaresta aina luomisesta tuhoamiseen. Concordia perustuu TCP/IP-protokolliin.

Edellä mainittu palvelinohjelmisto koostuu seuraavista osista [Mitsubishi, 1998]:

- *Agenttipalvelija* (agent manager) tarjoaa agenttien liikkumiseen tarvittavan perusrakenteen. Agenttipalvelija huolehtii agentin elinkaaresta.

- *Turvallisuuspalvelija* (security manager) suojaa agenteja sekä palvelimien resursseja. Turvallisuuspalvelijan vastuulla on myös käyttäjien ja heidän agenttiansa tunnistaminen.
- *Tallennuspalvelija* (persistence manager) pitää huolta, että agenttien tila säilyy agenttien siirtymisenkin aikana.
- *Kommunikointipalvelija* (event manager) huolehtii agenttien välisten viestien välittämisestä. Kommunikointivalvojan välityksellä agentit voivat myös synkronoida ja jakaa tietoa keskenään.
- *Siirtopalvelija* (queue manager) huolehtii agenttien luotettavasta siirtämisestä palvelinohjelmistojen välillä.
- *Nimipalvelija* (directory manager) tarjoaa nimipalvelun sekä sovelluksille että agenteille.
- *Hallinnointipalvelija* (administration manager) tarjoaa etähallinta ominaisuuden Concordia-järjestelmään. Hallinnointipalvelijan avulla voidaan samanaikaisesti hallita useita palvelinohjelmistoja.
- *Agenttien työkalukirjasto* (agent tool library) tarjoaa liikkuvien agenttien kehittämiseen tarvittavat ohjelmointirajapinnat ja agenttiluokat

Lisäksi Concordia-järjestelmään kuuluu niin sanottu *palvelusilta* (service bridge), jonka kautta ohjelmoija voi halutessaan lisätä ominaisuuksia palvelinohjelmistoon. Ohjelmoimalla palvelusiltoja voidaan palvelinohjelmiston ominaisuuksia lisätä ja laajentaa agenttien mahdollisia sovellusalueita.

4.6.5. Grasshopper

Grasshopperin [IKV++, 1998] kerrotaan olevan Java-perustainen agenttikehys ja ensimmäinen liikkuvien agenttien ympäristö, joka on yhteensopiva agenttien liikkuvuuden ja hallinnoinnin standardin, MASIFin kanssa.

Grasshopper-järjestelmässä palvelimella on *agenttuuri* (agency), jossa puolestaan voi olla yksi tai useampi *paikka* (place). *Agenttuurin perusosa* (core agency) sisältää vain agentin toiminnalle välttämättömimmät palvelut. Paikat puolestaan ovat kokoelma agenttuurin kehittyneempiä palveluita; yhteen paikkaan voi olla kerättynä esimerkiksi faksien tai sähköpostien lähettämiseen liittyviä palveluita. Järjestelmän agentit voivat vierailla sekä agenttuurin perusosassa että paikoissa.

Agentit vierailevat paikoissa käyttäkseen kyseisen paikan tarjoamia palveluita. Agenttuurin perusosassa agentit vierailevat saadakseen tietoa muista agenteista, agentuureista tai paikoista tai siirtyäkseen toiseen agenttuuriin.

Järjestelmää voidaan laajentaa ohjelmoimalla erityisiä *palveluagentteja* (service agents), jotka hoitavat jotakin tiettyä tehtävää.

Agenttuuri muodostuu seuraavista osista [IKV++, 1998]:

- *Rekisteröintipalvelun* (registration service) vastuulla on tietää agenttuurissa olevat paikat ja agentit.
- *Kommunikaatiopalvelu* (communication service) tarjoaa mahdollisuuden agenttien väliseen kommunikointiin useilla eri protokollilla.
- *Siirtopalvelu* (transport service) mahdollistaa agenttien siirtymisen toiseen agenttuuriin. Määränpäässään agentti jatkaa suoritustaan siitä kohdasta mihin se jäi ennen siirtoa.
- *Turvallisuuspalvelu* (security service) on kehitetty takaamaan agenttien turvallinen suoritus.
- *Hallinnointipalvelu* (management service) on kehitetty, jotta käyttäjä voisi seurata ja kontrolloida agenttuurin tapahtumia. Hallinnointipalvelun avulla käyttäjä voi hallita sekä agenttuurissa olevia agenteja että paikkoja.

4.6.6. JATLite

Heecheol [1998a] kuvaa JATLiteä Java-kieleen perustuvaksi järjestelmäksi, joka avulla voidaan kehittää TCP/IP-pohjaiseen kommunikointiin perustuvia hajautettuja agenttijärjestelmiä. JATLite-järjestelmän agentit ovat staattisia, mutta järjestelmän kommunikointiominaisuudet ovat edistyksellisiä. JATLiten kommunikointiominaisuudet mahdollistavat esimerkiksi applettien käytön agenteina. Vaikka JATLite keskittyy lähinnä *KQML-viestien* (Knowledge Query and Manipulation language) sekä *SMTP-* (Simple Mail Transfer Protocol) ja *FTP-protokollien* (File Transfer Protocol) välityksellä kommunikoiviin agentteihin, on myös muunlaisien viestien lähettäminen mahdollista.

JATLite muodostuu viidestä eri kerroksesta (layer) [Heecheol, 1998b]:

1. *Abstraktikerros* (abstract layer) on kerroksista alin. Abstraktikerros määrittelee abstraktit luokat, jotka ylemmät kerrokset toteuttavat.
2. *Kommunikointikerros* (base layer) huolehtii järjestelmän TCP/IP-pohjaisesta kommunikoinnista.
3. *KQML kerros* (KQML layer) luo tuen KQML-kielelle. Järjestelmän käyttäjien tulisi käyttää agenttien väliseen kommunikointiin KQML-viestejä. Järjestelmän agentit pystyvät kommunikoimaan myös muunlaisilla viesteillä, mutta tätä varten täytyy muokata KQML kerrosta.

4. *Reitityskerros* (router layer) reitittää agenttien lähettämiä viestejä. Reitityskerroksen avulla voidaan kiertää applettien tiedonsiirtorajoituksia, mikä mahdollistaa myös applettien käyttämisen agenteina.
5. *Protokollakerros* (protocol layer) tarjoaa agenteille mahdollisuuden käyttää SMTP- ja FTP-protollia kommunikoimiseen. Protokollakerroksesta on hyötyä etenkin applettina toimiville agenteille, jotka voivat tiedonsiirtorajoituksistaan huolimatta käyttää edellä mainittuja protokollia kommunikoimiseen.

Yksi JATLite-järjestelmän keskeisimmistä osista on *agenttireititin* (agent router), joka Heecheolin [1998a] mukaan mahdollistaa eri palvelimilla olevien agenttien kommunikoimisen. Myös appletit voivat toimia agenteina. Mikä tahansa järjestelmän agentti voi luoda yhteyden agenttireitittimeen, joka puolestaan välittää viestin vastaanottavalle agentille. Jos kyseessä on agenttina toimiva applet, täytyy agenttireitittimen sijaita samalla palvelimella, josta applet käynnistettiin. Käyttäessään agenttireititintä viestinvälitykseen agenttien ei tarvitse pitää kirjaa muiden agenttien sijainnista eikä luoda erillisiä yhteyksiä niihin.

4.6.7. Kehitysympäristöjen vertailua

Java-To-Go, Mole, Concordia, Grasshopper ja luvussa viisi kuvattava Aglets ovat luonteeltaan hyvin samankaltaisia järjestelmiä. Kaikissa järjestelmissä agentit pystyvät liikkumaan palvelimelta toiselle, kunhan palvelimessa on järjestelmän oma palvelinohjelmisto. Lukuunottamatta Java-To-Go:ta kaikissa järjestelmissä agentit pystyvät tallentamaan tilansa milloin tahansa ja jatkamaan suoritustaan toisella palvelimella. Järjestelmät tarjoavat mahdollisuuden paitsi agenttien liikkumiseen myös agenttien väliseen kommunikointiin.

Edellä esitellyistä agenttijärjestelmistä mielenkiintoisin viestinvälitysjärjestelmä oli JATLitessä. JATLiten viestinvälitysjärjestelmän agenttireititin mahdollistaa myös applettien toimimisen agenteina, pystyen jopa FTP-tiedonsiirtoihin millä tahansa palvelimella sijaitsevan toisen agentin kanssa.

Molessa, Grasshopperissa ja Concordiassa järjestelmän palvelinohjelmiston laajentaminen oli tehty erityisen helpoksi. Palvelinohjelmiston tarjoamia palveluita pystyi laajentamaan itse tehdyillä agenteilla. Grasshopperissa palvelinohjelmistolla oli mielenkiintoinen arkkitehtuuri, joka jakoi palvelinohjelmiston perusosaan ja kehittyneempiä palveluita tarjoaviin paikkoihin.

Kaikenkaikkiaan kaikki käsitellyt agentin liikkuvuuden mahdollistavat järjestelmät olivat luonteeltaan hyvin samanlaisia. Joissakin järjestelmissä oli jokin osa järjestelmää kehitetty pidemmälle kuin muissa.

Seuraava taulukko tiivistää edellä kuvattujen liikkuvien agenttien kehittämiseen tarkoitettujen kehitysympäristöjen muutamia keskeisimpiä ominaisuuksia. Taulukossa on esitetty vastaavat ominaisuudet myös Aglets-järjestelmästä, jota käsitellään tarkemmin luvussa viisi.

	Palvelimella sijaitseva alusta	Voiko agentti keskeyttää suorituksensa ja siirtyä toiselle palvelimelle milloin tahansa	Ominaisuuksien lisääminen
Java-To-Go	Palvelinohjelmisto (hall server)	ei	Palvelinohjelmiston laajennuksilla
Mole	Sijainti (location)	kyllä	Ohjelmoimalla resursseja hallinnoivia järjestelmäagentteja
Concordia	Palvelinohjelmisto (Concordia server)	kyllä	Palvelusillan avulla
Grasshopper	Paikka (place)	kyllä	Ohjelmoimalla palveluagentteja
Aglets	Agenttikonteksti	kyllä	Ohjelmoimalla tiettyyn tehtävään erikoistuneita agentteja

Taulukko 1. Eri liikkuvien agenttien kehitysympäristöjen keskeisimpien ominaisuuksien vertailua.

4.6.8. Muita kehitysympäristöjä

Erilaisia agenttien kehittämiseen tarkoitettuja järjestelmiä on toteutettu useita, edellä on niistä kuvattu tarkemmin vain muutamia hyvin dokumentoituja ja Java-kieleen perustuvia järjestelmiä. Seuraavassa esitellään lyhyesti muutamia muita agenttien kehittämiseen tarkoitettuja järjestelmiä.

Open Agent Architecture [SRI International, 1996] on agenttiarkkitehtuuri, jossa agenteista muodostetaan yhteistoiminnallisia yhteisöjä. Jotta agentti voisi olla osana tällaista yhteisöä sen tulee osata rekisteröidä tarjoamansa palvelut, kommunikoida *ICL-kielellä* (Interagent Communication Language) ja sen tulee omata kaikille OAA:n agenteille yhteinen perustoiminnallisuus. OAA-agentteja voidaan luoda monilla ohjelmointikielillä.

Intelligent Agent Library [Bits & Pixels, 1999] on Java-kieleen perustuva agenttikehys, jossa agentit voivat kommunikoida esimerkiksi KQML-kielellä. Intelligent Agent Library koostuu viidestä eri Java-pakkauksesta, joista kukin toteuttaa jonkin toiminnallisuuden.

IBM:n ABE-järjestelmä (Agent Building Environment) kehitys on jo lopetettu. ABE:n seuraajaksi on kehitetty Java- ja *XML-kieliin* (Extensible Markup Language) perustuva CommonRules [IBM, 1999b].

Myös Java Agent Template:n [Frost, 1996] kehittäminen on lopetettu ja sen on korvannut aiemmin esitelty JATLite.

JAFMAS:n [Baker, 1998] (A Java-based Agent Framework for Multi-Agent Systems) 16 Java-luokkaa luovat järjestelmän hajautettujen agenttijärjestelmien toteuttamiseen. JAFMAS-järjestelmän agentit pystyvät kommunikoimaan esimerkiksi KQML-kielen välityksellä. JAFMAS-järjestelmä keskittyy pääasiassa agenttien yhteistyön kehittämiseen moniagenttijärjestelmissä.

4.7. Yhteenveto

Suurin este liikkuvien agenttien suosion kasvamiselle on liikkuvien agenttien vaatimukset toimintaympäristölleen. Jotta agentit voisivat siirtyä palvelimelle, täytyy palvelimessa olla agentille soveltuva palvelinohjelmisto. Vaikka erilaisia palvelinohjelmistoja ja kehitysympäristöjä on kehitetty useita, ei mikään niistä ole saavuttanut kovin laajaa suosiota verrattuna esimerkiksi staattisiin agentteihin.

Käytettäessä liikkuvia agenteja järjestelmien toteuttamisessa voidaan siitä hyötyä esimerkiksi verkon kuorman vähenemisenä. Siirtyessään palvelimelle agentit voivat myöskin saada suuremmat oikeudet kuin toiselta palvelimelta yhteyden ottavat agentit.

5. Agletit

Yksi liikkuvien agenttien kehitysympäristö on IBM:n kehittämä Aglets-järjestelmä [IBM, 1999a]. Agletit ovat IBM:n kehittämia liikkuvia agentteja, jotka on toteutettu Java-kielellä. Tässä luvussa termillä agentti viitataan nimen omaan Aglets-järjestelmän agletteihin. Aglets-järjestelmä on erityisesti suunniteltu siten, että sen avulla on helppo toteuttaa erilaisia liikkuviin agentteihin perustuvia järjestelmiä. Näin ohjelmoija pääsee paremmin keskittymään itse sisällön tuottamiseen, eikä agenttien perusominaisuuksien, kuten liikkumisen ja viestinvälityksen, toteuttamiseen tarvitse kiinnittää huomiota. Aglets-järjestelmä on myös yhteensopiva aikaisemmin kuvatun MASIF-standardin kanssa.

Aglets-järjestelmä valittiin tarkemman tarkastelun kohteeksi ja myös oman järjestelmän toteutusarkkitehtuuriksi, koska se on yksi laajimmin käytössä olevista Javalla toteutetuista liikkuvien agenttien kehitysympäristöistä. Aglets-järjestelmä tarjoaa myös useita erilaisia tapoja agenttien väliseen viestinvälitykseen, mikä on erityisen tärkeää kehitettäessä moniagenttijärjestelmiä.

5.1. Agenttien vertailua appletteihin

Aglets-järjestelmän agentit perustuvat hieman samanlaiseen tekniikkaan kuin appletit: varsinaisen ohjelman luokkatiedosto siirretään suoritettavaksi toiseen paikkaan. Erona on kuitenkin se, että agentin mukana siirtyy myös ohjelman tila, toisin kuin appletissa [Venners, 1997]. Kun applet siirretään, siirtyy tällöin vain ohjelman koodi, agenttia siirrettäessä siirretään käynnissä oleva ohjelma eli koodin lisäksi myös ohjelman tila. Jos agentti on esimerkiksi suorittamassa jotakin laskutoimitusta, niin siirtyessään toiseen paikkaan se jatkaa laskutoimitusta siitä mihin se edellisessä paikassa jäi. Vastaavassa tilanteessa applet aloitaisi laskemisen uudelleen alusta.

Applettien ja agenttien siirtämisessä on myös muita eroavaisuuksia. Siirrettäessä appletteja tapahtuu siirto yleensä palvelimelta käyttäjän koneelle ja siirto on täysin riippuvainen käyttäjän toimista. Jos käyttäjä esimerkiksi siirtyy WWW-sivulle, jossa on applet, siirretään applet palvelimelta käyttäjän koneelle suoritettavaksi. Tämän jälkeen applet ei enää siirry, vaan suorittaa tehtävänsä tuolla sivulla.

Agentteja puolestaan siirretään yleensä useampia kertoja. Agenttien siirtäminen alkaa useimmiten siitä, että käyttäjä luo agentin omalla koneellaan ja siirtää sen palvelimelle. Tämän jälkeen agentti voi siirtyä itsenäisesti, käyttäjän toimista riippumatta, muille palvelimille agentin tilasta ja toiminnallisuudesta riippuen. Suoritettuaan tehtävänsä agentti voi siirtää itsensä takaisin käyttäjän koneelle.

Kuten applettikin myös agentti ajetaan omassa säikeessään (tai useassa). Ajoympäristönä agenteilla on niin sanottu *agenttikonteksti*, joka sijaitsee palvelinkoneella. Agenttikonteksti on hieman vastaava kuin selainohjelmien appletteja varten luoma Java-ympäristö, joka osaa ladata applettien luokkatiedostot verkosta [Venners, 1997]. Agentteja varten luotu agenttikonteksti osaa ladata verkosta sekä agenttien luokkatiedostoja että myös agenttien suoritus-tilan. Agentti ei siis voi siirtyä sellaiseen palvelimeen, jolla ei ole tällaista kontekstia. Yhdessä palvelimessa voi olla myös useita konteksteja. Jokaisella kontekstilla on myös turvallisuudesta vastaava olio, eräänlainen *turvallisuusohjeisto* (security manager), joka suojaa palvelinta. Vastaava turvallisuusohjeisto on myös selaimien Java-ympäristöissä, joissa ohjeisto suojaa vihamielisiltä tai väärin toimivilta appleteilta.

5.2. Agenttipalvelin

Jotta agentti voisi siirtyä tiettyyn koneeseen, pitää tässä koneessa olla *agenttipalvelin*. Aglets Software Development Kitin (ASDK) mukana tulee tämä palvelinohjelma. Yhdessä koneessa voi olla käynnissä useampiakin agenttipalvelimia, jos ne vain toimivat eri TCP-porteissa. Näin agentteja ja niiden liikkumista voidaan testata yhdessäkin koneessa ilman varsinaista verkkoyhteyttä. Koneessa pitää olla kuitenkin käytössä TCP/IP-protokollat, koska agenttien liikkuminen perustuu TCP/IP-protokollien päällä toimivaan *ATP-protokolla*n (Agent Transfer Protocol).

Agenttipalvelimen mukana tulee myös *agenttiselain* (aglet viewer), Tahiti. Agenttiselaimessa näytetään kaikki ne agentit, jotka ovat agenttipalvelimella. Kullakin agenttipalvelimella on oma agenttiselain, josta pystyy helposti seuraamaan miten agentit liikkuvat eri palvelimien välillä. Agenttiselaimen avulla pystyy helposti käsittelemään agentteja (luomaan, tuhoamaan, lähettämään eteenpäin, kutsumaan kotiin ja niin edelleen) sekä säätämään joitakin turvallisuusparametreja.

5.3. Agenttien ympäristö

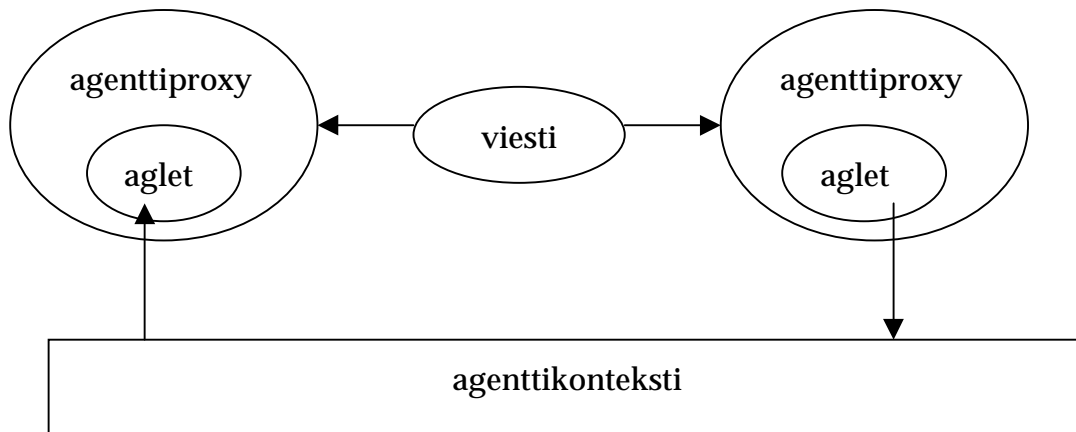
Kuten aikaisemmin todettiin, liikkuvat agentit eivät yksinään voi tehdä mitään, koska ne tarvitsevat palvelimella sijaitsevan tietynlaisen alustan. Aglets-järjestelmässä tämä alusta on agenttipalvelimen agenttikonteksti. Palvelimella pitää siis olla vähintään yksi agenttikonteksti, ennen kuin agentit voivat siirtyä palvelimeen. Agentit siis liikkuvat kontekstien välillä ja hankkivat tietoa palvelimesta kontekstin kautta. Kontekstin lisäksi keskeisiä käsitteitä Aglets-järjestelmässä ovat agenttiproxy ja viesti.

Agenttiproxy on eräänlainen agentin edustaja, jonka kautta agentin kanssa kommunikoidaan. Viestit ovat sanomia, joita välitetään agenttien välillä. Tarvemmin sanoen viestejä välitetään agenttiproxyjen välillä ja agenttiproxy välittää viestin edustamalleen agentille.

Koko elinkaarensa ajan agentilla on uniikki *tunniste* (identifier), jonka avulla agentit voidaan yksilöidä. Tunnistetta käytetään muun muassa pyydetessä agenttikontekstilta jonkin tietyn agentin agenttiproxy.

Lange ja Oshima [1998] kuvaavat agenttiproxyn tarkoitusta muutenkin kuin viestinvälityksessä. Agenttiproxy toimii agentin edustajana laajemminkin kuin vain viestinvälityksessä. Agenttiproxy toimii myös suojana, joka estää agentin julkisten metodien kutsumisen suoraan. Agenttiproxyn avulla voidaan myös piilottaa agentin todellinen sijainti: paikallinen agenttiproxy voi edustaa toisessa järjestelmässä olevaa agenttia.

Toisessa järjestelmässä olevaa agenttia edustava paikallinen agenttiproxy voidaan saada kahdella eri tavalla. Luodessa agentti saadaan sitä vastaava agenttiproxy. Kun agentti lähetetään toiseen järjestelmään, säilyy paikallinen proxy agentin edustajana. Toinen tapa on agentin agenttiproxyn noutaminen toisesta järjestelmästä. Agenttiproxyn noutamista varten täytyy tietää sekä agentin sijainti että sen tunniste, jotta voitaisiin yksilöllisesti määritellä minkä agentin agenttiproxyä ollaan noutamassa.



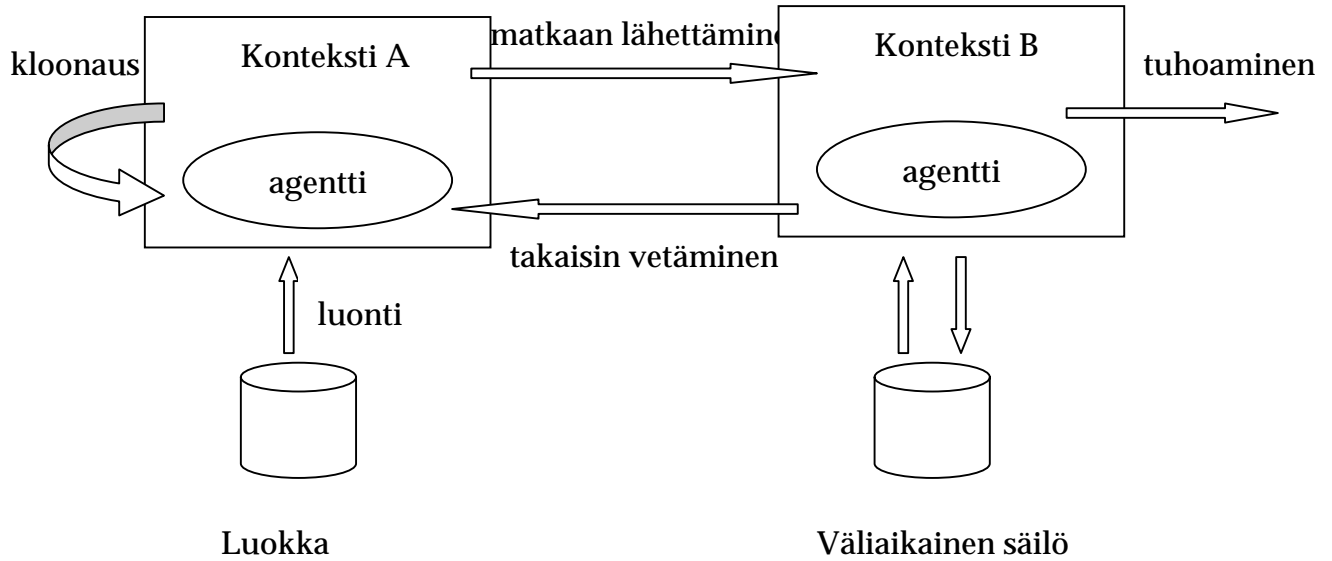
Kuva 2. Agenteille tulevat viestit välitetään agenteja edustavien agenttiproxyjen kautta. Agentit hankkivat tietoa palvelimesta agenttikontekstin kautta ja pystyvät siirtymään vain sellaiselle palvelimelle, jossa on agenttikonteksti. [Oshima et al, 1998]

5.4. Agenttien elinkaari

Elinkaarensa aikana agentti voi kokea useita erilaisia tapahtumia. Tällaisia tapahtumia ovat:

- luonti: uusi agentti luodaan, sen tila alustetaan ja säie käynnistetään,
- kloonaus: luodaan uusi agentti, joka on aikaisemman agentin kaltainen (myös aikaisemman agentin tila kopioituu),
- matkaan lähettäminen: agentti siirtyy toiselle palvelimelle,
- takaisin vetäminen: aikaisemmin matkaan lähetetty agentti vedetään takaisin toiselta palvelimelta,
- deaktivointi: agentti asetetaan odotustilaan, agentin suorituksen tila tallennetaan,
- aktivointi: aikaisemmin deaktivoitu agentti jatkaa toimintaansa levyltä palautettavasta tilasta, ja
- tuhoaminen: agentti tuhoetaan, agentin tila tuhoutuu lopullisesti.

Agentin luontia ja tuhoamista käsitellään tarkemmin kohdassa 5.5.1 ja agentin siirtymistä kohdassa 5.5.2.



Kuva 3. Agentti voi kokea elinkaarensa aikana useita erilaisia tapahtumia. [Oshima et al, 1998]

Lukuunottamatta tuhoamista, luontia ja kloonausta joudutaan kaikissa edellä mainituissa tapahtumissa agentin tila joko tallentamaan levyille tai siirtämään verkkoa pitkin. Agentin suorituksen tilan tallentamiseen tai siirtämiseen tarvitaan kumpaankin samaa tekniikkaa: *sarjallistamista* (serialization) [Venners, 1997].

Sarjallistaminen tarkoittaa sitä, että olio ja sen tila kirjoitetaan johonkin tietovirtaan (stream). Tämän virran kautta olio voidaan tallentaa levyille tai vaihtoehtoisesti vaikka lähettää TCP-portin kautta toiselle koneelle [Sun, 1997]. Sarjallistamista käytetään esimerkiksi Javan RMI-tekniikassa. Vastaavasti siirretty tai tallennettu olio voidaan myöhemmin palauttaa.

Jotta jokin olio ja sen tila voitaisiin sarjallistaa, pitää tämän olion olla ilmentymä luokasta, joka toteuttaa Javan rajapinnoista joko rajapinnan *Serializable* tai *Externalizable*. Näiden rajapintojen ero on siinä, että *Externalizable* rajapinnan toteuttavan luokan ilmentymä (olio) on itse vastuussa tietojen (datajäsenten) kirjoittamisesta virtaan. Myös rajapinnan *Serializable* toteuttavan luokan ilmentymä voi vaikuttaa siihen, mitä tietoja siitä kirjoitetaan virtaan; oletusarvoisesti kuitenkin kirjoitetaan kaikki datajäsenet.

5.5. Agentin käyttäytyminen

Agenttien toiminnan ja käyttäytymisen kannalta agenttien elinkaaren tapahtumat voidaan jakaa kolmeen eri ryhmään: luonti ja tuhoaminen, siirtyminen ja kommunikointi. Seuraavassa kuvataan lyhyesti kutakin näistä ryhmistä.

5.5.1. Luonti ja tuhoaminen

Lange ja Oshima [1998] kuvaavat agenttien luontia ja tuhoamista vaiheittain. Agentit luodaan aina jonkin palvelimen jossakin kontekstissa. Agentin luoja voi toimia joko jokin samassa kontekstissa oleva agentti tai kontekstin ulkopuolella oleva agentti tai jokin muu, jolla on riittävät oikeudet agentin luomiseen (esimerkiksi käyttäjä).

Agentti voidaan luoda joko instantioimalla uusi olio luokkamäärittelystä tai kloonamalla jo olemassa oleva agentti. Luominen koostuu seuraavista vaiheista [Lange ja Oshima, 1998]:

1. Olion luominen ja tunnisteen määrittäminen. Agentin luokkamäärittely ladataan ja luokasta luodaan olio, samalla oliolle annetaan yksilöllinen tunniste.
2. Alustaminen. Agentille annetaan mahdollisuus alustaa itsensä mahdollisesti saatujen alustusarvojen mukaan. Vasta alustuksen jälkeen agentti voi olettaa, että se on luotu oikein ja kokonaan.
3. Autonominen toimiminen. Kun agentti on varmistunut, että se on luotu oikein ja kokonaan, aloittaa se toimintansa. Agentti voi nyt toimia itsenäisesti olematta riippuvainen muista kontekstissa olevista agenteista (myös luojastaan).

Agentin tuhoamisen voi käynnistää edellä mainittujen mahdollisten luojien lisäksi myös agentti itse. Aglets-järjestelmä voi tuhota agentin esimerkiksi seuraavista syistä [Lange ja Oshima, 1998]:

1. Agentin elinaika päättyy: agentille määrätty elinaika on umpeutunut.
2. Agentille ei ole käyttöä: kukaan ei käytä agenttia tai ei viittaa siihen.
3. Turvallisuusohjeiden rikkominen.
4. Järjestelmän sulkeminen.

Lange ja Oshima [1998] kirjoittavat agentin tuhoamisen olevan kaksivaiheinen toimenpide. Aluksi agenttia varoitetaan tuhoamisesta ja sille annetaan

aikaa suorituksen loppuunsaattamiseen. Tämän jälkeen järjestelmä keskeyttää agentin suorituksen ja tuhoaa sen.

5.5.2. Siirtyminen

Langen ja Oshiman [1998] mukaan agentin siirtymisen voi käynnistää joko agentti itse, samassa kontekstissa oleva toinen agentti tai kontekstin ulkopuolella oleva agentti tai jokin muu riittävät oikeudet omaava osapuoli. Agentti lähetetään sen nykyisestä kontekstista ja vastaanotetaan johonkin toiseen kontekstiin. Vastaanottava konteksti voi myös kieltäytyä vastaanottamasta agenttia.

Kun agentti valmistautuu matkaan, on sen tunnistettava määränpäänä oleva järjestelmä. Kun kohde on tunnistettu, ilmoittaa agentti paikalliselle järjestelmälle, että se haluaa siirtyä kohdejärjestelmään [Lange ja Oshima, 1998]. Agentin määränpäänä oleva kohdejärjestelmä yksilöidään *URL-osoitteella* (uniform resource locator).

Kun paikallinen järjestelmä vastaanottaa agentin pyynnön siirtymisestä, toimii se seuraavasti [Lange ja Oshima, 1998]:

1. Agentin suorituksen keskeyttäminen. Agenttia varoitetaan siirtymisestä ja sille annetaan aikaa valmistautua siirtymiseen. Tämän jälkeen agentin säie pysäytetään.
2. Agentin sarjallistaminen. Agentin tila ja luokka sarjallistetaan, jotta se voitaisiin siirtää kohdejärjestelmään.
3. Sarjallistetun agentin koodaaminen. Sarjallistettu agentti koodataan käytetyn siirtoprotokollan mukaisesti.
4. Agentin siirtäminen. Järjestelmä luo yhteyden kohdejärjestelmään ja siirtää koodatun agentin.

Agentin siirtyminen käynnistetään useimpien agenttia koskevien tapahtumien tapaan agenttiproxyn kautta. Agenttiproxyn dispatch-metodille välitetään osoite, johon agentti halutaan siirrettävän. Tämän jälkeen käynnistyy agentin lähettäminen. Useimmiten agentin ensimmäisen siirtymisen alkuunpanee käyttäjä joko suoraan tai välillisesti. Käyttäjä voi tämän jälkeen kutsua lähettämänsä agentin takaisin, mutta useimmat agentit päättävät itse milloin ne palaavat tai vaihtoehtoisesti siirtyvät eteenpäin.

Käyttäjä voi huomata agentin lähettämisen agenttiselaimesta. Lähetettävä agentti häviää kontekstissa olevien agenttien listasta ja agenttiselaimen tilariville tulee ilmoitus agentin lähettämisestä.

Ennen kuin vastaanottava järjestelmä voi vastaanottaa siirtyvän agentin, tulee sen tietää, voiko se hyväksyä agentteja lähettävältä järjestelmältä [Lange ja Oshima, 1998]. Agentin siirto tapahtuu vasta sen jälkeen, kun agentin lähettävä järjestelmä on luotettavasti tunnistettu.

Agentin vastaanottaminen tapahtuu seuraavasti [Lange ja Oshima, 1998]:

1. Agentin vastaanotto. Kun vastaanottava järjestelmä on todennut, että se voi hyväksyä agentteja lähettävältä järjestelmältä, tapahtuu varsinainen agentin vastaanotto eli koodatun agentin siirtyminen järjestelmään.
2. Agentin koodauksen purku. Järjestelmä purkaa saapuvan tietovirran koodauksen.
3. Agentin palauttaminen. Agentti palautetaan sarjallistetusta tilasta, agentin luokka instantioidaan ja agentin tila palautetaan.
4. Suorituksen jatkaminen. Siirretylle agentille ilmoitetaan siirron valmistumisesta. Agentin suoritusta varten varataan oma säie ja agentti voi jatkaa suoritustaan.

Vastaavasti kuin agentin lähettämisen käyttäjä voi huomata myös agentin vastaanottamisen agenttiselaimesta. Vastaanotettu agentti ilmestyy kontekstissa olevien agenttien listaan ja agenttiselaimen tilariville tulee ilmoitus agentin vastaanottamisesta.

5.5.3. Kommunikointi

Lange ja Oshima [1998] kuvaavat yleisesti agenttien kommunikointia. Agentti voi kommunikoida sekä samassa järjestelmässä että muissa järjestelmissä olevien agenttien kanssa. Omatessaan riittävät oikeudet agentti voi kutsua toisen agentin jotakin metodia tai lähettää viestin toiselle agentille. Agentit voivat lähettää viestejä joko yhdelle tietylle agentille (peer-to-peer) tai joukolle agentteja (broadcast).

Jokaisella agentilla on oma *viestinvälittäjä* (message manager), joka asettaa saapuvat viestit jonoon. Pääsääntöisesti viestit ovat jonossa siinä järjestyksessä, kun ne ovat saapuneet. Agentti voi kuitenkin määrätä tietynlaisille viesteille joko korkeamman tai normaalia matalamman prioriteetin. Laittaessaan viestejä jonoon viestinkäsittelijä järjestää viestit niin, että saapuva viesti menee jonossa kaikkien alemmalla prioriteetilla olevien viestien ohi.

Agenttien välinen kommunikointi voidaan jakaa kolmeen eri ryhmään viestinvälitystavan mukaan: synkroniseen, asynkroniseen ja yksisuuntaiseen viestinvälitykseen. Näiden lisäksi agentit voivat kommunikoida myös

kontekstin välityksellä. Agentti voi tallentaa kontekstiin avain-arvo-pareja, joita mikä tahansa kontekstissa oleva agentti voi lukea turvallisuusasetusten niin salliessa. Vaikkei avain-arvo-parien tallennus suoranaisesti ole kommunikointia, voivat agentit välittää tietoa myös tällä tavalla. Esimerkiksi tiedonhaku-agentit voivat tallentaa kontekstiin tiedon siitä, että tässä kontekstissa on jo käyty. Vastaavasti agentti voi jättää jälkeensä avain-arvo-pareja ikäänkuin jalanjäljiksi, jotta se seuraavan kerran ollessaan samassa kontekstissa tietäisi lähteä "eri suuntaan" kuin edellisellä kerralla.

Langen ja Oshiman [1998] mukaan *synkroninen viestinvälitys* on yleisimmin käytetty viestinvälitystyyppi. Synkronisessa viestinvälityksessä viestin lähettävä agentti voi jatkaa suoritustaan vasta sen jälkeen, kun viestin vastaanottava agentti on käsitellyt viestin ja vastannut siihen.

Jos järjestelmässä on paljon agenteja, jotka lähettävät viestejä toisilleen, voi synkroninen viestinvälitys keskeyttää viestin lähettävän agentin suorituksen pitkäksi aikaa. Etenkin, jos agentti on lähettänyt sellaisen viestin, jolle vastaanottava agentti on määrännyt normaalia pienemmän prioriteetin. Tällöin viesti voi olla vastaanottavan agentin viestijonossa pitkäänkin, ennen kuin se käsitellään ja viestin lähettänyt agentti saa vastauksen.

Asynkronisessa viestinvälityksessä viestin lähettänyt agentti voi jatkaa suoritustaan myös sillä välin, kun viestin vastaanottanut agentti käsittelee lähetettyä viestiä. Lähettävä agentti saa viittauksen (handle), jota käyttäen se voi myöhemmin lukea vastauksen [Lange ja Oshima, 1998]. Asynkroninen viestinvälitys on joustava tapa agenttien väliseen kommunikointiin, koska viestin lähettävän agentin ei tarvitse odottaa vastaanottavan agentin vastausta, vaan se voi jatkaa suoritustaan. Asynkroninen viestinvälitys on erityisen hyödyllinen useiden agenttien kommunikoidessa keskenään.

Myös *yksisuuntainen viestinvälitys* on luonteeltaan asynkroninen eli se ei keskeytä viestin lähettävän agentin suoritusta. Yksisuuntaisessa viestinvälityksessä viestin lähettänyt agentti ei saa viittausta vastaukseen, eikä viestin vastaanottavan agentin ole pakko vastata viestiin [Lange ja Oshima, 1998]. Tästä viestinvälitystyyppistä käytetään joskus myös nimitystä "ammu ja unohda".

5.6. Agenteista saatava hyöty elektronisille kauppapaikoille

Aikaisemmin kuvattiin elektronisten kauppapaikkojen yleisimpiä ongelmia. Seuraavaksi pohditaan, voitaisiinko agenteilla ratkaista joitakin näistä ongelmista tai helpottaako agenttien käyttö ohjelmoijan työtä näiden ongelmien ratkaisemisessa. Erityisesti keskitytään Aglets-järjestelmän ominaisuuksiin ja niiden mahdollistamiin ratkaisuihin.

5.6.1. Turvallisuus

Wong *et al* [1999] kuvailevat Java-ohjelmointikielen, jolla Aglets-järjestelmäkin on toteutettu, turvallisuusohjeiston merkitystä liikkuville agenttijärjestelmille. Java-ohjelmointikieleen on toteutettu turvallisuusohjeisto, jota noudattaen järjestelmä voi suojata itseään. Kaikki muut liikkuvan agenttijärjestelmän osat kommunikoivat turvallisuusohjeiston kanssa tunnistaakseen ja suojatakseen agentteja. Turvallisuusohjeisto voi esimerkiksi suojella agentteja käyttämällä salausta ennen agenttien siirtämistä tai tallentamista.

Turvallisuusohjeiston avulla käyttäjä voi antaa agenteille tavallista suurempia oikeuksia, esimerkiksi oikeuden lukea ja/tai kirjoittaa johonkin tiettyyn tiedostoon. Halutessaan käyttäjä voi myös "sammuttaa" turvallisuusohjeiston, jolloin agentit voivat muun muassa lukea minkä tahansa käyttäjän koneella olevan tiedoston tai kirjoittaa siihen.

Agenttikonteksti on vastuussa siinä vierailevista agenteista, jotka puolestaan pystyvät hankkimaan tietoa palvelimesta kontekstin kautta. Langen ja Oshiman [1998] mukaan Aglets SDK:n seuraavassa versiossa¹ (Aglets SDK 1.1) on mahdollista asettaa kontekstikohtaisia turvallisuusohjeistoja, jotka valvovat, millaisia tietoja agentit voivat kunkin kontekstin kautta hankkia (esimerkiksi pääsy tiedostojärjestelmään, verkkoyhteydet, ynnä muut sellaiset). Yhdellä palvelimella voi olla samaan aikaan useita eri konteksteja, ja näillä kaikilla voi olla erilainen turvallisuusohjeisto niiden käyttötarkoituksesta riippuen.

5.6.2. Resurssien valvonta

Aglets-järjestelmä ei nykyisellään tue prosessorin ja muistin resurssien valvontaa. Pelkästään agenttien määrän valvonta on kuitenkin helppo toteuttaa järjestelmään erityisillä hallinnointiagenteilla, jotka pitävät kirjaa järjestelmässä olevista agenteista. Agentti voidaan ohjelmoida siten, että saapuessaan kauppapaikkaan ja lähtiessään sieltä se ilmoittaa siirtymisistään erityiselle hallinnointiagentille, joka toimii tilanteen vaatimalla tavalla. Jos kauppapaikka on "täynnä", voi hallinnointiagentti käännä takaisin kauppapaikkaan tulevat agentit, kunnes jokin kauppapaikassa ollut agentti siirtyy toiselle palvelimelle

¹ Viitattaessa Aglets SDK:n seuraavaan versioon viitataan Aglets SDK 1.1:een. Tekstissä kuvatut ominaisuudet perustuvat kuitenkin Aglets SDK 1.1 Beta 1 version turvallisuusmalliin, joka voi vielä muuttua [Lange ja Oshima, 1998].

ja kauppapaikassa on jälleen tilaa uusille agenteille. Elektronisen kauppapaikan ylläpitäjä voi siis rajoittaa kauppapaikassa samaan aikaan olevien agenttien määrää.

Langen ja Oshiman [1998] mukaan Aglets SDK:n seuraavassa versiossa on mahdollista, että agenttikonteksti voi rajoittaa agentin käyttämää muistin määrää tai suorittimen kuormittamista. Aglets SDK 1.1:ssä myös vierailevien agenttien määrän rajoittaminen on lisätty osaksi itse järjestelmää. Lisäksi voidaan rajoittaa esimerkiksi myös agentin luomien kloonien määrää tai agentin elinaikaa.

5.6.3. Käytön valvonta

Käyttäjä voidaan tunnistaa agentin *käyttäjätunnisteen* (owner id) avulla. Kaikilla yhden käyttäjän agenteilla on sama käyttäjätunniste. Aglets SDK:n seuraavassa versiossa on vastaava käyttäjätunniste (aglet owner), jonka avulla voidaan tunnistaa agentin lähettäjä. *Valmistajatunnisteen* (aglet manufacturer) avulla puolestaan voidaan tunnistaa agentin implementoija [Lange ja Oshima, 1998]. Valmistajatunnisteen avulla voidaan varmistua, että agentti ei ole vihamielinen ja että agentti toimii oikein.

Jokaisella agentilla on myös oma yksilöllinen tunnisteensa. Jos agentti kaupankäynnin jossain vaiheessa siirtyy kotipalvelimelleen hakeakseen käyttäjältä vahvistuksen, voi kauppakumppani yksilöllisen tunnisteensa avulla varmistua siitä, että agentin palatessa kauppapaikkaan kyseessä on sama agentti.

Agentteihin pohjautuvassa järjestelmässä voidaan siis agentit todeta saman käyttäjän luomiksi agentin käyttäjätunnisteen avulla. Vaikka käyttäjätunniste ei olekaan yksilöllinen vaan käyttäjän itsensä valitsema, vaikeuttaa se huomattavasti toisena henkilönä esiintymistä järjestelmässä, joka seuraa myös käyttäjätunnisteita. Koska käyttäjätunniste on käyttäjälle täysin näkymätön, voidaan järjestelmää kehitettäessä yhdistää käyttäjätunnisteeseen vielä käyttäjätunnus ja salasana. Näiden kolmen tunnisteensa avulla käyttäjät voidaan tunnistaa ja käyttöä valvoa varsin luotettavasti.

5.6.4. Maksaminen

Aglets-järjestelmä, kuten kaikki muutkin yleiset agenttijärjestelmät, on kehitetty nimenomaan agenttien toteuttamiseen, eivätkä ne tue erityisesti maksamista elektronisen kaupankäynnin yhteydessä. Maksaminen on elektroniselle kaupankäynnille spesifinen ominaisuus, ja jotta jokin agenttijärjestelmä tukisi erityisesti juuri tätä ominaisuutta, tulisi agenttijärjestelmän olla kehitetty

nimenomaan elektronista kaupankäyntiä varten. Tällä hetkellä mikään yleisesti käytössä oleva agenttijärjestelmä ei tue maksamista elektronisessa kaupankäynnissä. Maksaminen on yksi elektronisen kaupankäynnin kriittisimmistä vaiheista, joten se tulisi hoitaa yleisten standardien mukaisesti.

Agentit voitaisiin ehkä ohjelmoida siten, että ne veloittavat käyttäjän omalla koneella sijaitsevasta elektronisesta lompakosta laskun suuruisen summan ja kuljettavat sen mukanaan elektroniselle kauppapaikalle. Tällainen maksujärjestely olisi suhteellisen turvallinenkin, koska agentti veloittaisi elektronista lompakkoa, joka sijaitsee samalla koneella kuin millä agentti on luotu. Lompakon veloittamiseen tarvittaisiin tietysti myös käyttäjän hyväksyntä (mahdollinen lompakon salasana). Muutoin agentteihin perustuvan elektronisen kauppapaikan täytyykin ainakin tällä hetkellä tyytyä yleisesti sähköisessä kaupankäynnissä käytössä oleviin maksuratkaisuihin.

5.7. Yhteenveto

Aglets-järjestelmä on Javalla toteutettu liikkuvien agenttien kehittämiseen tarkoitettu järjestelmä, jossa agentit liikkuvat palvelimilla sijaitsevien agenttikontekstien välillä hieman samaan tapaan kuin appletit siirtyvät palvelimelta selaimen Java-ympäristöön suoritettavaksi. Applettien tapaan agentit voivat myös avata erilaisia ikkunoita.

Aglets-järjestelmän agenteja voidaan jo nykyisinkin käyttää elektronisten kauppapaikkojen ongelmien ratkaisemiseen. Tulevaisuudessa järjestelmään ollaan vielä lisäämässä ominaisuuksia, joilla pystytään entistä paremmin ratkaisemaan joitakin elektronisten kauppapaikkojen ongelmia. Tällaisia piirteitä ovat muiden muassa erilaiset käyttäjä- ja valmistajatunnisteet sekä mahdollisuudet rajoittaa muistin ja suorittimen kuormitusta.

6. Oma järjestelmä

Esimerkkinä liikkuvilla agenteilla toteutetusta elektronisesta kauppapaikasta kehitin osana tätä työtä *EMMA-järjestelmän* (Experimental Marketplace with Mobile Agents). EMMA-järjestelmä on tarkoitettu käytettäväksi kuluttajalta kuluttajalle -tyyppisessä kaupassa esimerkiksi käytettyjen tavaroiden kauppamiseen. Järjestelmä vastaa siis ainakin osittain perinteistä kirpputoria.

Jotta järjestelmä pysyisi hieman yksinkertaisempana, eivät käyttäjät voi vapaasti valita, mitä tuotteita he kauppapaikassa myyvät tai ostavat. Koemiellessä ylläpidettävässä kauppapaikassa kauppatavarat on rajattu vain GSM-matkapuhelmiin. Järjestelmää voidaan kuitenkin helposti laajentaa koskemaan myös muita kauppatavaroita lisäämällä siihen myyntikategorioita. Kauppapaikan ylläpitäjä päättää millaisella myyntivalikoimalla hänen kauppapaikassaan voidaan käydä kauppaa ja tarpeen tullen laajentaa tai supistaa valikoimaa.

On myös mahdollista, että vain muutamiin kauppatavaroihin erikoistuvat kauppapaikat tulevat sekalaisia tavaroita kauppavia kauppapaikkoja suositummiksi kauppapaikkojen yleistyessä. Voi siis syntyä eräänlaisia erikoiskauppapaikkoja, joiden tunnettavuus leviää niiden hyvän maineen myötä. Myös tällöin myyntivalikoiman rajaaminen on tarpeellista. Rajatun myyntivalikoiman kauppapaikka on myös järjestelmänä kevyempi kuin vapaan myyntivalikoiman kauppapaikka. Vapaan myyntivalikoiman kauppapaikassa on ongelmana myös se, että miten kahden eri käyttäjän kirjoittama kuvaus samanlaisesta tavarasta osataan yhdistää.

Järjestelmän kehittämisen tarkoituksena on ollut osoittaa, että elektronisen kauppapaikan pystyy toteuttamaan pohjautuen vain liikkuviin agentteihin. Koska järjestelmä on kokeellinen ja resurssit tämän työn puitteissa varsin rajalliset, järjestelmä ei ole niin laaja, että se voisi sellaisenaan soveltua kovin laajamittaiseen käyttöön.

Kehitettäessä EMMA-järjestelmää on ollut tarkoitus toteuttaa kuluttajalta kuluttajalle -tyyppinen palvelu. Näin ollen järjestelmän pääasiallinen tarkoitus on tukea käyttäjää kauppakumppanin etsimisessä. Koska kauppaa käyvät kaksi kuluttajaa, maksujärjestelmän luominen järjestelmään ei ole järkevää, koska maksujärjestelyt voivat olla hyvinkin joustavat. Kuluttajalta kuluttajalle -tyyppinen kaupankäynti on usein käytettyjen esineiden kauppaa, joten ostaja haluaa todennäköisesti nähdä myytävän tuotteen ennen ostamista. Näin ollen ei ole järkevää luoda hankalaa maksujärjestelmää järjestelmän osaksi, koska ostaja todennäköisimmin maksaa ostamansa tavaran vasta nähtyään sen.

Maksujärjestelmän toteuttaminen ei olisi ollut mahdollistakaan tämän työn puitteissa kehittämisen vaatimien resurssien vuoksi.

Järjestelmä toteutettiin kokeellisesti liikkuvilla agenteilla ja se perustuu Aglets-arkkitehtuuriin. Tässä luvussa termillä agentti viitataan Aglets-järjestelmän agletteihin. Käytettäessä liikkuvia agenteja staattisten agenttien sijaan voidaan esimerkiksi vähentää verkon kuormaa. Verkon kuormaa pienennetään myös sillä, että järjestelmän käyttöliittymää mahdollisine suurine kuvineen ei ladata verkosta vaan se on liitetty kiinteästi itse agenttiin.

Seuraavassa kuvataan järjestelmän arkkitehtuuria ja järjestelmässä tarvittavia agenteja hieman tarkemmin.

6.1. Arkkitehtuuri

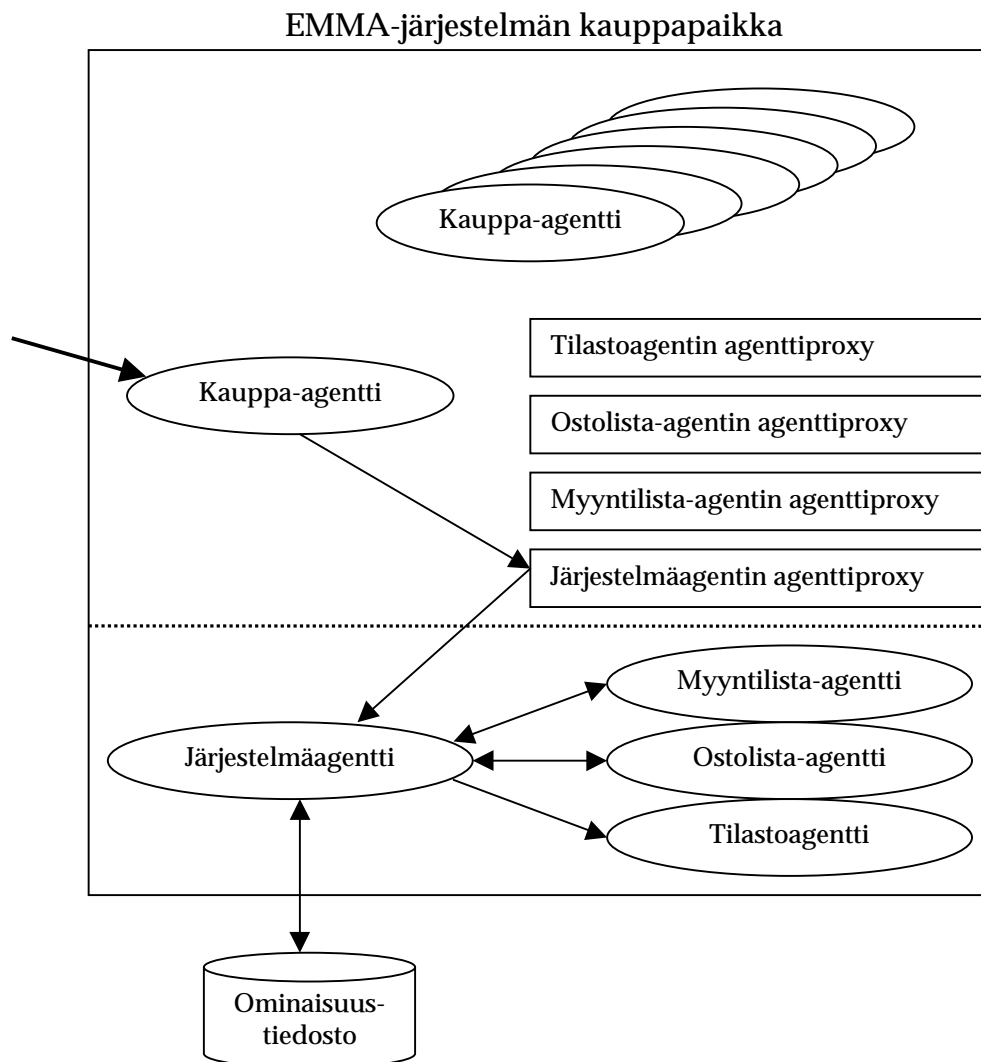
Järjestelmä koostuu erilaisista agenteista, jotka vaativat toimiakseen agenttipalvelimen. Kaikki järjestelmän toiminnot hoidetaan agenteilla, jotka voidaan jakaa kahteen eri ryhmään: hallinnolliset agentit ja kauppa käyvät agentit. Hallinnollisia agenteja voi olla kussakin kauppapaikassa vain yksi kappale kuttakin, kun taas kauppa käyviä agenteja voi olla useita. Eri agenttien ominaisuuksista kerrotaan tarkemmin kohdassa 6.2.

Järjestelmässä käytetään agenttien väliseen viestinvälitykseen kahta erilaista tapaa: sekä normaalia viestinvälitystä että tiedon välittämistä agenttikontekstiin talletettavien arvojen kautta. Normaalin viestinvälityksen kautta agentit vaihtavat keskenään tietoja silloin, kun on selvää kenelle viesti on tarkoitettu. Esimerkiksi kaksi kauppa käyvää agenttia välittävät keskenään kauppavaroihin liittyvää tietoa. Hallinnointiagentin ja kauppa käyvän agentin välillä liikkuu lähinnä juuri tämän tietyn agentin toimintaa ohjailevia viestejä.

Agenttikontekstiin talletettavia arvoja käytetään yleensä toisten agenttien paikallistamiseen. Kun kauppa käyvä agentti saapuu kauppapaikkaan, löytää se hallinnointiagentit kauppapaikan agenttikontekstiin tallennettujen arvojen perusteella. Normaalin viestinvälityksen ja agenttikontekstiin tallennettujen arvojen hyväksikäyttöä tarkastellaan tarkemmin kohdassa 6.2. kunkin agentin kohdalla erikseen.

Kullakin kauppapaikalla on omat ominaisuutensa, jotka ylläpitäjä määrittelee *ominaisuustiedostossa* (configuration file). Ominaisuustiedostossa on määriteltä kauppaapaikan nimi, maksimimäärä yhtäaikaaisesti kauppapaikassa oleville agenteille ja kauppapaikan myyntivalikoima. Jotta järjestelmä voisi lukea ominaisuustiedoston sisällön, täytyy kauppapaikan ylläpitäjän kuitenkin määrittellä kyseiselle tiedostolle lukuoikeudet. Oletusarvoisesti agenteilla ei ole luku- tai kirjoitusoikeuksia mihinkään palvelimen tiedostoon.

Ominaisuustiedostossa määrätyn agenttien lukumäärän ylärajan tarkoituksena on antaa kauppapaikan ylläpitäjälle mahdollisuus rajoittaa kauppapaikassa yhtä aikaa olevien agenttien määrää. Näin ylläpitäjä voi samalla rajata myös kauppapaikan kuluttamia resursseja. Jos kauppapaikka on täynnä, käännetään sinne tulevat agentit takaisin. Uusia agenteja kauppapaikkaan päästetään sitten, kun sieltä on aluksi poistunut jokin aikaisemmin sinne tullut agentti. Käyttäjä määrää kauppaa tekeville agenteille *voimassaoloajan* (validity time), jonka umpeuduttua agentit viimeistään poistuvat kauppapaikasta. Käyttäjä voi poistaa omat agenttinsa kauppapaikasta myös aikaisemmin.



Kuva 4. Kauppapaikan arkkitehtuuri. Kauppapaikkaan saapuva kauppa-agentti saa hallinnointiagenttien agenttiproxyt käyttöönsä agenttikontekstiin tallennetuista arvoista. Agenttiproxyjen kautta kauppa-agentti pystyy lähettämään viestejä hallinnointiagenteille, esimerkiksi järjestelmäagentille. Tarvittavat tiedot järjestelmästä järjestelmäagentti saa ominaisuustiedostosta.

EMMA-järjestelmässä kauppapaikan ylläpitäjä voi rajoittaa myyntivalikoimaa, jolla hänen kauppapaikassaan käydään kauppaa. Ylläpitäjä voi kirjoittaa ominaisuustiedostoon määritelmät haluamistaan myyntikategorioista ja kussakin kategoriassa olevista kauppatavaroista. Vain ominaisuustiedostossa määritellyistä kauppatavaroista voidaan käydä kauppaa kauppapaikassa.

Ennen kuin käyttäjä voi lähettää agenttinsa kauppapaikkaan, on agentin otettava yhteys kauppapaikkaan ja tiedusteltava mitä kauppatavaroita kauppapaikassa myydään. Jos käyttäjän haluamalla kauppatavaralla ei voida käydä kauppaa juuri tässä kauppapaikassa, voi käyttäjä kokeilla jotakin toista kauppapaikkaa. Käyttäjä ei siis pysty lähettämään kauppapaikkaan agenttia, joka yrittäisi myydä tai ostaa jonkin sellaisen tavarat, jota kauppapaikassa ei saa kaupata. Koemielessä ylläpidettävässä järjestelmässä myyntivalikoima on rajattu kolmen valmistajan GSM-matkapuhelimiin.

Jotta kauppapaikan myyntivalikoimaa voitaisiin rajata, täytyy siis sallitut kauppatavarat nimetä etukäteen. Sallittujen kauppatavaroiden nimeämisestä on erityisesti hyötyä myös silloin, kun kahden eri agentin kauppatavaroita verrataan keskenään. Jos kauppatavaroita ei olisi nimetty etukäteen, voisi kauppapaikassa olla myytävänä ja ostettavana samaa kauppatavaraa useallakin eri nimellä. Tällöin kahden eri agentin kauppatavaroiden yhtäläisyyden vertaaminen olisi erittäin hankalaa. Koska kauppatavarat on nimetty etukäteen, ja agenttien kauppaama tavara on aina jokin tässä listassa olevista tavaroista, on myyntitavaroiden vertaaminen helpompaa. Näin käyttäjän lähettämältä agentilta ei jää huomaamatta, jos jollakin toisella kauppapaikassa olevalla agentilla on kauppatavaranaan agentin omaa kauppatavaraa vastaava tuote.

Rajatulla myyntivalikoimalla voidaan hillitä myös mahdollista varastetun tavarat kauppaa. Elektronisella kirpputorilla epärehellisellä käyttäjällä voi olla pienempi mahdollisuus kiinnijäämiseen verrattuna esimerkiksi oikeaan kirpputoriin tai lehti-ilmoitukseen, joten elektroninen kirpputori voisi olla varsin houkutteleva kauppapaikka epärehelliselle käyttäjälle. Epärehellinen käyttäjä ei kuitenkaan pysty myymään kauppapaikassa muita kuin ominaisuustiedostossa määriteltäviä tavaroita. Kutakin myytävää tavaraa kohden käyttäjän on lisäksi luotava oma agenttinsa. Koska agenttien määrä kauppapaikassa on rajattu, vie jokainen tavaraa myyvä agentti paikan myös tavaraa ostavalta agentilta, joten kauppapaikat eivät sovellu esimerkiksi laajamittaiseen varastetun tavarat myymiseen. Myytäessä yksittäisiä turhiksi tulleita esineitä kauppapaikka toimii parhaimmillaan.

6.2. Järjestelmässä tarvittavia agentteja

Varsinainen kauppapaikka on agenttipalvelin, jossa on kauppapaikan ylläpitoon tarvittavat agentit eli hallinnointiagentit. Hallinnointiagentteja on neljä: järjestelmäagentti, tilasto agentti sekä myynti- ja ostolista-agentit. Hallinnointiagenttien toiminnasta kerrotaan tarkemmin kohdassa 6.2.1.

Varsinaisten hallinnointiagenttien lisäksi järjestelmään kuuluu yksi hallinnollinen agentti, jonka rooli on varsin poikkeava verrattuna muihin hallinnollisiin agentteihin. Järjestelmän luomiseen ja käynnistämiseen käytetään erityistä luontiagenttia. Luontiagentin tehtävänä on luoda tarvittavat hallinnointiagentit kauppapaikan ylläpitäjän agenttipalvelimelle. Näin ylläpitäjän ei tarvitse kauppapaikkaa käynnistäessään luoda kuin yksi agentti, joka puolestaan luo tarvittavat hallinnointiagentit. Hoidettuaan tehtävänsä luontiagentti tuhoaa itsensä.

Lisäksi järjestelmässä tarvitaan tietysti myös varsinaiset kauppaa tekevät agentit. Kauppaa tekeviä agentteja on kolme: myyntiagentti, ostoagentti ja seililuagentti. Lisäksi joillakin agenteilla on erityisiä apuriagentteja.

6.2.1. Hallinnointiagentit

Hallinnointiagentteja on olemassa neljä: järjestelmäagentti, tilastoagentti, myyntilista-agentti ja ostolista-agentti. Kutakin näistä voi olla yhdessä kauppapaikassa vain yksi kappale. Hallinnointiagentteja vastaavia toimintoja ei oikeissa kirpputoreissa varsinaisesti ole. Lähimpänä hallinnointiagenttien toimenkuvaa oikeissa kirpputoreissa olisi tilat vuokraava henkilö, joka myöskin pitää tilastotietoja siitä, kuinka paljon myyjiä ja ostajia hänen tiloissaan on käynyt.

Kaikille hallinnointiagenteille on yhteistä se, että niitä voi olla kussakin kauppapaikassa vain yksi kappale. Kaikissa hallinnointiagenteissa on myös samanlainen mekanismi useiden samankaltaisten agenttien luomisen estämiseksi. Mekanismissa käytetään hyväksi agenttikontekstiin tallennettavia arvoja. Kun hallinnointiagenttia luodaan, tarkistetaan agenttikontekstista hallinnointiagentin tyyppiä vastaava arvo, joka kertoo kuinka monta tämän kaltaista agenttia kontekstissa on jo olemassa. Jos arvoa ei löydy tai se on nolla, tarkoittaa se, että kontekstissa ei ole vielä tämän tyyppistä hallinnointiagenttia. Tällöin agentti luodaan normaalisti. Jos arvo kuitenkin on jotain muuta, tuhotaan hallinnointiagentti välittömästi sen luonnin jälkeen. Jos nyt luotu hallinnointiagentti oli ensimmäinen laatuaan, tallentaa se arvon yksi agenttikontekstin hallinnointiagentin tyyppiä vastaavaan arvoon. Näin mahdollisesti seuraavana luotu samantyyppinen hallinnointiagentti tietää kaltaisansa olemassaolosta. Kun arvon

asettanut hallinnointiagentti lopulta tuhotaan, käy se asettamassa arvon nol-
laan. Näin voidaan välttää useiden samanlaisten hallinnointiagenttien luomi-
nen samaan agenttikontekstiin.

Toinen kaikille hallinnointiagenteille yhteinen piirre on se, että niitä ei
voida lähettää toiselle palvelimelle. Näin ollen kauppapaikka voidaan perustaa
vain omassa käytössä olevalle palvelimelle. Hallinnointiagenttien lähettämisen
estäminen on toteutettu toteuttamalla liikkumista valvova kuuntelijaolio, joka
ei salli agentin siirtämistä.

Järjestelmän keskeisin agentti on järjestelmäagentti. Järjestelmäagentti
huolehtii esimerkiksi siitä, että kauppapaikassa ei ole enempää kauppaa teke-
viä agenteja kuin ylläpitäjä on määrännyt. Kun kauppaa tekevä agentti saapuu
kauppapaikkaan, ottaa se ensimmäisenä yhteyttä järjestelmäagenttiin. Jos
kauppapaikka on jo täynnä, järjestelmäagentti käännäyttää kauppaa tekevän
agentin takaisin palvelimelle, josta se tuli. Jos kauppapaikassa on vielä tilaa, il-
moittaa järjestelmäagentti uuden agentin saapumisesta joko myyntilista- tai os-
tolista-agentille saapuneen agentin tehtävästä riippuen. Lisäksi järjestelmä-
agentti ilmoittaa agentin saapumisesta myös tilastoagentille riippumatta agen-
tin tehtävästä. Vastaavasti agentit ilmoittavat järjestelmäagenttia poistumises-
taan.

Jotta kauppapaikkaan tulevat agentit löytäisivät järjestelmäagentin, on
järjestelmäagentin agenttiproxy tallennettu agenttikontekstiin tietyllä, kaikkien
agenttien tuntemalla arvolla. Jokainen kontekstiin tuleva ja arvon tunteva
agentti voi siis saada haltuunsa järjestelmäagentin agenttiproxyn. Agentti-
proxyn avulla kauppapaikkaan tulevat agentit voivat kommunikoida järjestel-
mäagentin kanssa.

Myös muiden hallinnointiagenttien agenttiproxyt on tallennettu agentti-
kontekstiin. Kun kauppapaikkaa ollaan luomassa, luodaan aluksi kaikki muut
hallinnointiagentit ja onnistuneiden luontien päätteeksi tallennetaan niiden
agenttiproxyt agenttikontekstiin. Viimeisenä hallinnointiagenttina luodaan jär-
jestelmäagentti, joka tarkistaa löytyykö agenttikontekstista kaikkien hallinnoin-
tiagenttien agenttiproxyt. Näin voidaan varmistua siitä, että järjestelmä tulee
luotua oikein ja kauppapaikassa on kaikki tarvittavat agentit. Samalla järjestel-
mäagentti saa käyttöönsä tarvitsemansa muiden hallinnointiagenttien agentti-
proxyt.

Järjestelmäagentti on myös vastuussa rajatusta myyntivalikoimasta. Järjes-
telmäagentti on ainut hallinnointiagentti, jolla on hallussaan lista kauppapai-
kassa sallituista kauppavararoista. Ennen varsinaista kauppapaikkaan siirty-
mistä kauppaa tekevä agentti ottaa yhteyttä kauppapaikkaan kysyäkseen mil-
lainen myyntivalikoima tässä kauppapaikassa on. Järjestelmäagentti käsittelee

tämän pyynnön ja antaa sallitun myyntivalikoiman kauppaa tekevälle agentille.

Osto- ja myyntilista-agenttien toiminta on hyvin samankaltaista. Ostolista-agentti pitää kirjaa kauppapaikassa olevista ostoagenteista ja myyntilista-agentti myyntiagenteista. Kauppaa tekevän agentin saapuessa lista-agentit saavat ilmoituksen tapahtumasta järjestelmäagentin kautta. Kauppaa tekevää agenttia vastaava lista-agentti tallentaa saapuneen agentin agenttiproxyn listaan, jonka jälkeen saapuneelle agentille annetaan lupa aloittaa kaupankäynti. Lupa annetaan sen jälkeen, kun saapunut agentti on saatu rekisteröityä kauppapaikkaan oikein. Rekisteröinnillä tässä tarkoitetaan järjestelmäagentin ylläpitämien agenttien lukumäärän päivittämistä ja saapuneen agentin agenttiproxyn lisäämistä lista-agentin listaan.

Vastaavasti agentin poistuessa kauppapaikasta lista-agentit saavat tapahtumasta ilmoituksen järjestelmäagentilta ja poistavat poistuvan agentin agenttiproxyn listasta. Lista-agenttien vastuulla on tarjota saapuville myyntiagenteille lista kauppapaikassa olevista ostoagenteista ja vastaavasti saapuville ostoagenteille lista kauppapaikassa olevista myyntiagenteista.

Myös lista-agenttien agenttiproxyt on tallennettu agenttikontekstiin. Lista-agenttien agenttiproxyt on tallennettu agenttikontekstiin, jotta myynti- tai ostoagentit voisivat helposti lähettää viestejä lista-agenteille. Myynti- ja ostoagentit tiedustelevat lista-agentilta kauppapaikassa olevien mahdollisten kaupakumppaniensa agenttiproxyja. Lista-agenttien agenttiproxyt on tallennettu agenttikontekstiin myös järjestelmäagenttia varten, joka tarvitsee muiden hallinnointiagenttien agenttiproxyjä ja tarkistaa niiden olemassaolon luontinsa yhteydessä.

Hallinnointiagenteista vähiten tärkeä kaupankäynnin kannalta on tilastoagentti. Tilastoagentin tehtävänä on vain pitää kirjaa siitä kuinka monta agenttia kauppapaikassa on kaikenkaikkiaan käynyt kauppapaikan käynnistämisen jälkeen. Tilastoagentti pitää kirjaa sekä agenttien yhteismäärästä että erityyppisten agenttien määrästä. Tilastoagentti palvelee lähinnä kauppapaikan ylläpitäjää, joka haluaa seurata kuinka paljon liikennettä hänen kauppapaikassaan on ollut. Myös tilastoagentin agenttiproxy on tallennettu agenttikontekstiin järjestelmäagenttia varten, aivan kuten muidenkin hallinnointiagenttien.

Tiedot saapuvista agenteista tilastoagentti saa järjestelmäagentilta, joka lähettää tilastoagentille viestin aina agentin saapuessa kauppapaikkaan. Järjestelmäagentin tilastoagentille tekemää ilmoitusta saapuneesta agentista ei lasketa varsinaiseen saapuvan agentin rekisteröitymiseen. Agentti voidaan rekisteröidä kauppapaikkaan oikein, vaikkei tieto saapuneesta agentista jostain syystä tavoittaisikaan tilastoagenttia.

Järjestelmäagentti on ainoa hallinnointiagentti, jolla on varsinainen käyttöliittymä eli dialogi. Muut hallinnointiagentit osaavat vastata ylläpitäjän agenttipalvelimen kautta antamaan dialog-pyyntöön kirjoittamalla ylläpitäjälle hyödylliset tiedot komentoriville Java-konsoliin. Muiden agenttien antamat tiedot ovat tietoja agenttien lukumäärästä. Tilastoagentti kirjoittaa kaikkien kauppapaikassa käyneiden agenttien lukumäärän, kun taas lista-agentit kirjoittavat kumpikin omassa listassaan olevien agenttien lukumäärän. Järjestelmäagentti tiedustelee dialogissaan ylläpitäjältä ominaisuustiedoston sijaintia. Ylläpitäjän annettua ominaisuustiedoston sijainnin järjestelmäagentti näyttää dialogissa tiedostossa olleet sallitut kategoriat ja kauppatavarat sekä agenttien enimmäismäärän kauppapaikassa.

6.2.2. Kauppaa käyvät agentit

Kauppaa käyviä agentteja on olemassa kolme: myyntiagentti, ostoagentti ja selailuagentti. Kauppapaikassa voi olla samaan aikaan useita eri kauppaa käyviä agentteja. Itse asiassa kauppapaikan toiminta perustuu nimenomaan siihen, että kauppapaikassa on samaan aikaan useita eri myynti- ja ostoagentteja, jotka kommunikoivat keskenään ja vertailevat myyntiartikkeleitaan ja niiden hintoja.

Toisin kuin hallinnointiagenteille kauppaa käyville agenteille löytyy vastaavuudet oikeista kirpputoreista. Kirputoreilla on sekä tavaraa myyviä että ostavia henkilöitä sekä lisäksi henkilöitä, jotka ovat vain katselemassa mitä kaikkea kirpputorilla on myytävänä.

Yksinkertaisin kauppaa tekevästä agenteista on selailuagentti. Selailuagentin yksinkertaisuuden voi huomata jo agentin dialogista, jossa kysytään pelkästään kohteena olevan kauppapaikan osoite. Käyttäjän annettua kauppapaikan osoitteen selailuagentti siirtyy kauppapaikkaan ja ilmoittautuu järjestelmäagentille. Selailuagenttia ei kuitenkaan lisätä lista-agenttien listalle, koska selailuagentin kanssa ei voi varsinaisesti käydä kauppaa.

Selailuagentti noutaa myyntilista-agentin agenttiproxyn agenttikontekstista ja lähettää tälle viestin, johon myyntilista-agentti vastaa lähettämällä takaisin kaikkien kauppapaikassa olevien myyntiagenttien agenttiproxyt. Tämän jälkeen selailuagentti käy läpi kaikki myyntiagentit, kysyy niiltä niiden myymien tuotteiden tiedot, tallentaa ne ja palaa kotipalvelimelleen. Palattuaan kotipalvelimelleen selailuagentti esittää tiedot käyttäjälle. Selailuagentti siis kävi kauppapaikassa selaamassa mitä tuotteita siellä on myytävänä.

Myyntiagentti ja ostoagentti ovat toiminnaltaan hyvin samankaltaiset. Agentin luonnin jälkeen agentti näyttää käyttäjälle dialogin, jossa kysytään kohteena olevan kauppapaikan osoitetta. Saatuaan osoitteen agentti luo

apuriagentin, jonka se lähettää käyttäjän antamaan osoitteeseen. Apuriagentti siirtyy palvelimelle ja ilmoittautuu järjestelmäagentille, jolta apuriagentti saa kauppapaikan myyntivalikoiman. Tämän jälkeen apuriagentti palaa kotipalvelimelleen ja antaa myyntivalikoiman isäntäagentilleen, jotta tämä puolestaan voisi näyttää myyntivalikoiman dialogissaan.

Palatessaan takaisin apuriagentin pitäisi tunnistaa isäntäagenttinsa, jotta voisi antaa tälle kauppapaikan järjestelmäagentilta saamansa myyntivalikoima. Luodessaan apuriagentin isäntäagentti tallentaa oman agenttiproxynsä agenttikontekstiin. Agenttiproxy tallennetaan agenttikontekstiin avaimenaan apuriagentin tunniste. Kun apuriagentti palaa kotipalvelimelleen, hakee se agenttikontekstista isäntäagentin agenttiproxyn käyttäen omaa yksilöllistä tunnistettaan avaimena. Agenttiproxyn saamisen jälkeen apuriagentti antaa myyntivalikoiman isäntäagentille, jonka jälkeen apuriagentti tuhoataan.

Isäntäagentti näyttää nyt dialogissaan käyttäjälle kauppapaikan myyntivalikoiman. Käyttäjän tulee valita aluksi sopiva kategoria suunnittelemansa myynti- tai ostokohteen mukaan. Valittuaan kategorian käyttäjä valitsee tuosta kategoriasta myytäväksi tai ostettavaksi suunnitellun kauppatavaran. Kun kauppatavara on valittu, pitää käyttäjän antaa agentille yhteystietonsa. Yhteystietojen muoto on täysin vapaa, mutta koska ne ovat ainoat yhteystiedot, jotka välittyvät kaupankäynnin toiselle osapuolelle, tulisi niiden olla riittävän kattavat tätä tarkoitusta silmällä pitäen.

Yhteystietojen antamisen jälkeen käyttäjän tulee antaa agentille voimassaoloaika ja kauppatavaran osto- tai myyntihinta. Agentin voimassaoloaika annetaan tunteina ja hinta dollareina. Kun kaikki kentät on täytetty, on agentti valmis siirtymään kauppapaikkaan.

Kun agentti on siirtynyt kauppapaikkaan, ottaa se yhteyden järjestelmäagenttiin. Saatuaan luvan kaupankäynnin aloittamiseen, ottaa agentti yhteyden lista-agenttiin, jolta se saa listan kauppapaikassa olevien mahdollisten kauppakumppaneiden agenttiproxyista. Tämän jälkeen agentti alkaa vertailemaan kauppatavaraansa ja sen hintaa kauppakumppaniensa kanssa. Jos kauppakumppanilla on sopiva kauppatavara, tallentavat molemmat kauppakumppanit toistensa tiedot. Näin agentti käy läpi kaikki kauppapaikassa olevat mahdolliset kauppakumppanit. Tämän jälkeen agentti jää odottamaan kauppapaikkaan tulevia mahdollisia kauppakumppaneita, jotka puolestaan ottaisivat siihen yhteyttä vastaavalla tavalla.

Saapuessaan kauppapaikkaan luo agentti itselleen ajastimen. Kun agentti ollut kauppapaikassa käyttäjän antaman voimassaoloajan, ajastin laukeaa ja agentti yrittää palata takaisin kotiin. Jos kotiinpaluu ei kuitenkaan onnistu esimerkiksi sen vuoksi, että kotipalvelimen agenttipalvelin ei ole käynnissä,

asettaa agentti ajastimen uudelleen käyntiin. Tällä kertaa ajastin asetetaan kahdeksi tunniksi, jonka jälkeen paluuta yritetään uudelleen. Kaikenkaikkiaan agentti yrittää paluuta kymmenen kertaa. Jos vielä tällöinkään paluu ei onnistu, agentti tuhotaan, koska käyttäjä on luultavimmin unohtanut agentin. Koekäytössä olleessa kauppapaikassa käytettiin ajastimen aikayksikkönä tuntien sijaan sekunteja, jotta järjestelmää olisi helpompi testata.

Palattuaan kotipalvelimelle agentti näyttää keräämänsä tiedot käyttäjälle. Käyttäjä näkee kustakin kriteereitään vastanneesta osto- tai myynti-ilmoituksesta vastaavat tiedot kuin hän kirjoitti omasta kauppatavarastaan. Lisäksi agentti näyttää erillisessä listassa tiedot sellaisista ilmoituksista, jotka vastasivat vain kauppatavaraa käyttäjän ilmoituksessa. Tässä listassa hinta ei siis vastannut käyttäjän asettamia kriteereitä. Tämän listan tarkoituksena on palvella käyttäjää sellaisissa tapauksissa, että yhtään varsinaista kriteereitä vastaavaa osto- tai myynti-ilmoitusta ei löytynyt.

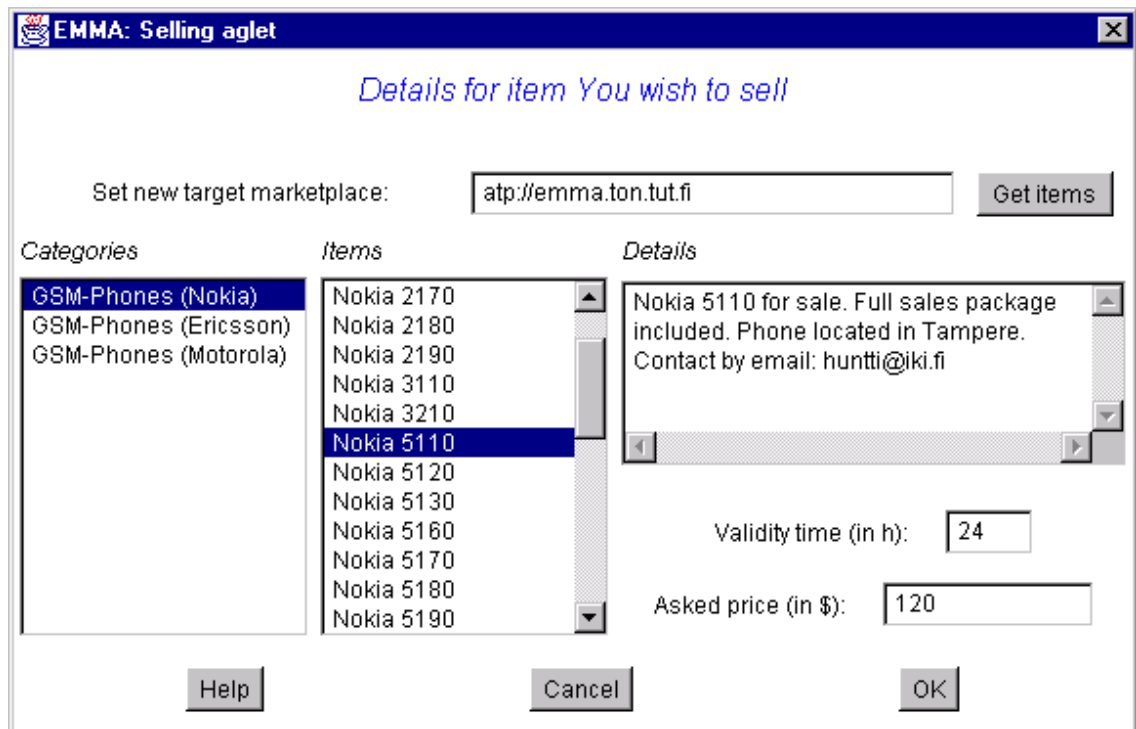
6.2.3. Muut agentit ja oliot

Edellä kuvattujen agenttien lisäksi järjestelmään kuuluu useita erilaisia apuri- ja muita agentteja tai muita olioita. Esimerkiksi sekä myynti- että ostoagentilla on kummallakin oma apuriagenttinsa, joka noutaa kauppapaikasta myyntivalikoiman. Lisäksi kummallakin on omat agenttidialoginsa, joissa tiedustellaan tietoja kauppatavarasta, virhedialoginsa, joissa näytetään käyttäjälle tieto tapahtuneesta virheestä sekä tulosdialoginsa, joissa näytetään käyttäjälle agentin keräämät tulokset. Lisäksi on olemassa ohjedialogi, jossa voidaan näyttää käyttäjälle tarkoitettuja ohjeita. Tämä dialogiluokka on yhteinen sekä järjestelmäagentille, osto- ja myyntiagentille että selailuagentille.

Näiden lisäksi järjestelmään kuuluu erilaisia kuuntelijaolioita sekä liikkuvuustapahtumia, näppäinpainalluksia että erilaisissa listoissa tapahtuneita tapahtumia varten.

6.3. Järjestelmän käyttöliittymä

Järjestelmän käyttöliittymä muodostuu agenttien käyttäjälle näyttämistä dialogeista. Agentti siis piirtää kaikki järjestelmästä käyttäjälle näkyvät osat, eikä lataa verkosta esimerkiksi kuvia tai HTML-tiedostoja. Tässä suhteessa EMMA-järjestelmä eroaa aikaisemmista agenteilla toteutetuista elektronisista kauppapaikoista. Aikaisemmissa järjestelmissä on järjestelmän käyttöliittymä ollut yleensä HTML-sivu, jonka käyttäjä on ladannut selaimellaan.



Kuva 5. Myyntiagentin käyttöliittymä. Kauppapaikka, johon käyttäjä aikoo agentin lähettää, on osoitteessa atp://emma.ton.tut.fi. Kauppapaikasta on noudettu kauppapaikan myyntivalikoima, jonka jälkeen käyttäjä on valinnut myytävän kohteen ja täyttänyt muut yksityiskohdat. Vastaavanlainen käyttöliittymä on myös ostoagentilla.

Liitettäessä järjestelmän käyttöliittymä kiinteästi itse agenttiin saavutetaan selainriippumattomuus. Myös käyttäjän huomaamat vasteajat ovat pienemmät kuin WWW:stä ladatulla käyttöliittymällä. Ratkaisun huonona puolena on, että kaikilla EMMA-järjestelmää käyttävillä elektronisilla kauppapaikoilla on oletusarvoisesti samanlainen käyttöliittymä. Käyttöliittymää voidaan kylläkin muokata päivittämällä käyttäjälle näkyvät agentit, mutta järjestelmän ylläpitäjä ei voi päättää milloin muuttaa järjestelmänsä käyttöliittymää; kunkin käyttäjän käyttöliittymä vaihtuu silloin, kun käyttäjä päivittää käyttämänsä agentin.

Järjestelmään kyllä olisi mahdollista tehdä myös WWW-pohjainen käyttöliittymä toteuttamalla HTTP-pyyntöjä käsittelevä agentti. Yksinkertainen esimerkki tällaisesta agentista on toteutettu myös Aglets SDK:n asennuspaketissa. Toteutettaessa järjestelmä kokonaan Aglets-järjestelmään pohjautuen saavutettiin kuitenkin etua myös toteutuksen yksinkertaistumisena. Jos esimerkiksi järjestelmän käyttöliittymä olisi ollut HTML-sivu, olisi järjestelmään jouduttu liittämään varsin monimutkainen HTTP-pyyntöjä käsittelevä agentti. Jotta useat eri käyttäjät olisivat voineet käyttää järjestelmää samanaikaisesti, olisi tarvittu

myös salasanat käyttäjien tunnistamiseen sekä useita muita mahdollisesti monimutkaisiakin ominaisuuksia, joita nykyisessä toteutuksessa ei tarvittu.

Ongelmana selaimen kautta käytettävässä kauppapaikassa olisi ollut myös se, että millä tavalla agentti olisi ilmoittanut käyttäjälle suorittaneensa tehtävän. Tällaisessa järjestelmässä käyttäjän täytyisi itse muistaa käydä tarkistamassa millaisia tuloksia agentti on saanut. Toteutetussa ratkaisussa agentti ilmoittaa käyttäjälle tulokset palattuaan takaisin käyttäjän koneelle.

6.4. Toteutuksen arvointia ja vertailua

Elektronisen kauppapaikan toteuttaminen liikkuvilla agenteilla onnistui mielestäni suhteellisen hyvin. Tavoitteena oli toteuttaa pelkästään liikkuviin agentteihin perustuva elektroninen kauppapaikka ja tämä tavoite saavutettiin.

Toteutettu kauppapaikka toteuttaa kohdassa 2.1 mainituista elektronisen kauppapaikan tarkoituksista [Chouldhury *et al* 1998] parhaiten ensimmäisen eli potentiaalisten kauppakumppaneiden tunnistamisen. Lisäksi järjestelmä tukee käyttäjää toisena mainitussa tarkoituksessa eli tietyn kauppakumppanin valinnassa. Vakkei järjestelmä suoraan valitsekaan yhtä tiettyä kauppakumppania, tukee järjestelmä käyttäjää tämän päätöksen tekemisessä.

Toteutuksen aikana esiin tuli lukuisia ongelmia, joista vaikeimmat liittyivät Aglets-järjestelmän turvallisuusasetuksiin. Turvallisuusasetukset rajoittavat esimerkiksi agenttien oikeutta lukea agenttikontekstiin tallennettuja arvoja. Vaihtoehtoinen tapa arvojen tallentamiselle agenttikontekstiin olisi ollut tiettyjen agenttien kirjautuminen kuuntelemaan tiettyjä viestejä.

Tällaisessa ratkaisussa esimerkiksi järjestelmäagentti olisi kirjautunut kuuntelemaan esimerkiksi viestiä "järjestelmäagentti" ja vastannut siihen lähettämällä oman agenttiproxynsä. Kauppapaikkaan saapuva agentti olisi lähettänyt viestin "järjestelmäagentti", jolloin kaikki tuota viestiä kuuntelemaan kirjautuneet olisivat saaneet tuon viestin. Tässä tapauksessahan kuuntelijoita olisi ollut vain yksi. Tämän ratkaisun huonona puolena on se, että jos kauppapaikkaan saapuu useita agentteja lähes samanaikaisesti erilaisia viestejä lähetetään paljon. Kaikki agentit ottavat yhteyttä järjestelmäagenttiin, jonka jälkeen agentit ottavat yhteyttä osto- tai myyntilista-agenttiin. Käyttämällä agenttikontekstia agenttiproxyjen välittämiseen välttyttiin turhien viestien lähettämiseltä.

Toteutuksen aikana tuli pienenä yllätyksenä se, että agenttien siirtyminen agenttikontekstista toiselle kestää niinkin kauan. Jos agenttikontekstit ovat samalla koneella, on siirtyminen nopeaa, koska kaikki tarvittavat luokat ovat jo valmiiksi myös vastaanottavan agenttikontekstin saatavilla. Siirtyminen toisella koneella olevaan agenttikontekstiin kuitenkin kestää yllättävän kauan.

Toteutusta on vaikea vertailla muiden elektronisten kauppapaikkojen toteutukseen, koska vastaavan järjestelmän olemassaolosta ei ole tietoa. Useimmiten elektroniset kauppapaikat ovat jollakin WWW-palvelimella sijaitsevia ja ovat kauppiaalta kuluttajalle -tyyppisiä kauppapaikkoja. Tällaisissa kauppapaikoissa voi kyllä olla varsin kehittyneet hakujärjestelmät, jotta käyttäjät varmasti löytäisivät haluamansa, mutta tällaisissa järjestelmissä käyttäjä ei voi asettaa omia tuotteitaan myyntiin.

Lähimpänä toteutetun järjestelmän toteutusta lienee Kasbah-järjestelmä [MIT, 1997], joka sekin on tarkoitettu yksittäisten tavaroiden myymiseen. Kasbahin käyttöliittymä on HTML-pohjainen, mutta varsinainen kaupankäminen tapahtuu agenteilla. Kasbahissa agentit tosin luodaan itse kauppapaikassa eikä käyttäjän koneella.

Myös e-Marketplace [IBM, 1997b] on hieman samankaltainen kuin toteutettu järjestelmä. e-Marketplace on kuitenkin sovelluskehys eikä varsinainen kauppajärjestelmä. e-Marketplacen käyttöliittymä voi olla HTML-pohjainen tai se voi olla erillinen asiakasohjelmisto. Koska e-Marketplace on kaupallinen sovelluskehys on siinä kuitenkin paljon sellaisia ominaisuuksia, jotka puuttuvat toteutetusta järjestelmästä. Esimerkkejä tällaisista ominaisuuksista ovat useiden eri kauppapaikkojen yhteistoiminta ja erityiset mainosagentit.

Jos toteutettua järjestelmää alettaisiin laajentaa, yritettäisiin siihen varmastikin saada joitakin samanlaisia piirteitä kuin e-Marketplacessa. Etenkin useiden kauppapaikkojen yhteistoiminnallisuus olisi käyttäjän kannalta erittäin hyödyllinen ominaisuus. Muodostamalla kauppapaikkojen verkosto saataisiin käyttäjälle mahdollisimman suuri hyöty pienellä vaivalla. Tällä hetkellä järjestelmässä ei ole varsinaista agenttien välistä neuvottelua, vaan agentit pelkäävät vertailevat niille annettuja toimeksiantoja. Kehitettäessä järjestelmää olisi-kin agenttien välisen todellisen neuvottelun toteuttaminen yksi kehitystoimenpiteistä.

Koska agentteihin pohjautuvan järjestelmän käyttöliittymä (eli siis agenttien käyttöliittymä) helpostikin jää hieman karummaksi kuin WWW-pohjaisen järjestelmän käyttöliittymä, olisikin yksi mielenkiintoinen jatkokehityksen kohde erityisten käyttöliittymämodulien kehittäminen. Agenttien ja käyttöliittymämodulien välille määriteltäisiin rajapinta, jonka jälkeen kuka tahansa voisi toteuttaa käyttöliittymämodulin joka olisi yhteensopiva järjestelmän kanssa toteuttaessaan määritellyn rajapinnan. Käyttäjä voisi vaihtaa agenttijärjestelmän käyttöliittymän ulkoasua ottamalla käyttöön uuden käyttöliittymämodulin.

7. Pohdintaa

Kehitettäessä elektronisia kauppapaikkoja tulisi kiinnittää huomiota elektronisten kauppapaikkojen kaupankäyntiprosessin kokonaisvaltaiseen tukemiseen. Usein elektroninen kauppapaikka toteuttaa hyvin yhden tai muutaman kaupankäynnin osa-alueen, mutta myös muihin osa-alueisiin tulisi kiinnittää huomiota. Esimerkiksi maksujärjestelyissä ja käyttäjän tunnistamisessa olisi paljon parannettavaa.

Käyttäjän tunnistamisen perustuminen käyttäjätunnukseen ja salasanaan on yleinen, mutta ei välttämättä kovin hyvä tapa. Useat käyttäjät valitsevat kuitenkin salasanoiden aivan liian helppoja sanoja, esimerkiksi saman sanan kuin käyttäjätunnukseen. Maksamisen tulisi olla elektronisessa kauppapaikassa mahdollisesti jopa helpompaa kuin perinteisen kaupan kassalla. Useissa kauppapaikoissa maksaminen on jo suhteellisen helppoa: maksaminen voi tapahtua esimerkiksi Meritan Solo-palvelulla ostosten yhteydessä. Kuitenkin esimerkiksi myös Solo-palvelu perustuu pankilta saatuun käyttäjätunnukseen ja salasanoihin, jotka vaihtuvat jokaisella käyttökerralla. Niinpä käyttäjä ei voi tehdä ostoksiaan muualla kuin kotona kuljettamatta mukanaan salasanalistaa.

Liikkuviin agentteihin perustuvissa elektronisissa kauppapaikoissa maksamisen saaminen osaksi järjestelmää on erityisen hankalaa, koska tällä hetkellä ei ole olemassa sellaista liikkuvien agenttien kehitysympäristöä, joka tukisi myös maksamista elektronisessa kaupankäynnissä. Jotta kaikki kaupankäynnin vaiheet voitaisiin saada osaksi liikkuvilla agenteilla toteutettua elektronista kauppapaikkaa, tulisi kehittää nimenomaan elektroniseen kaupankäyntiin soveltuva liikkuvien agenttien järjestelmä.

Lisäksi liikkuviin agentteihin pohjautuvien järjestelmien laajamittaista käyttöä jossakin yleisessä tarkoituksessa hidastaa liikkuvien agenttien tarvitsemat erilliset palvelinohjelmistot. Järjestelmien käytön tulisi olla mahdollisimman helppoa käyttäjille; palvelinohjelmiston asennus voi olla liian monimutkainen ja vaivalloinen tehtävä monelle tietokoneen käyttäjälle.

Mikään tällä hetkellä olemassaolevista liikkuvien agenttien kehitysympäristöistä tuskin tulee yleistymään elektronisten kauppapaikkojen toteuttamisessa, koska järjestelmät on kehitetty yleisiksi liikkuvien agenttien kehitysympäristöiksi eikä niissä ole tukea elektroniselle kaupankäynnille ominaisille piirteille. Koska elektronisessa kaupankäynnissä on useita sellaisia vaiheita, joiden tulee olla luotettavia sekä ostajan että myyjän kannalta, eivät tällaiset

yleiset agenttijärjestelmät ainakaan vielä tällä hetkellä kykene tarjoamaan tarvittavia ominaisuuksia.

Jotta esimerkiksi maksujärjestelyt saataisiin tehtyä käyttäjälle mahdollisimman joustavaksi, täytyy kuitenkin aluksi pystyä luotettavasti yksilöimään ja tunnistamaan käyttäjät. Maailmanlaajuista luotettavaa käyttäjien yksilöintiä on vaikea kehittää, mutta lähitulevaisuudessa Suomessa käyttöönotettavat henkilökohtaiset elektroniset henkilökortit voivat ratkaista tämän ongelman suomalaisissa kauppapaikoissa.

Maksamista kehitettäessä täytyy kuitenkin ottaa huomioon elektronisen kauppapaikan luonne, esimerkiksi elektronisessa kirpputorissa monimutkaisten maksujärjestelmien toteuttaminen ei ole järkevää. Perinteisten kauppojen tapaan elektronisen kauppapaikan tulisi kiinnittää huomiota myös kauppaa tukeviin toimintoihin, kuten esimerkiksi asiakaspalautteeseen ja tuotetukeen. Elektronisen kauppapaikan toimintaa voidaan tukea myös ilmaisilla oheistoiminnoilla, esimerkiksi joillakin viihteellisillä palveluilla.

Liikkuvien agenttien käyttö elektronisen kauppapaikan toteuttamisessa tulee kuitenkin luultavasti muuttamaan nykyistä käsitystä elektronisesta kauppapaikasta. Kauppapaikka ei enää välttämättä olekaan vain WWW:ssä oleva sivusto, jonka kautta voi tilata erilaisia tuotteita, vaan elektroninen kauppapaikka siirtyy osittain agenttien myötä myös käyttäjän koneelle. Esimerkiksi osana tätä työtä toteuttamassani EMMA-järjestelmässä käyttäjän tarvitsemat agentit luodaan käyttäjän omalla koneella, josta ne sitten siirtyvät varsinaiseen kauppapaikkaan. Suoritettuaan tehtävänsä agentit palaavat takaisin käyttäjän koneelle.

Paitsi elektronisten kauppapaikkojen (niin kuin muidenkin elektronisten palveluiden) luonne myös niiden käyttötavat voivat muuttua lähitulevaisuudessa etenkin langattomien tiedonsiirtotapojen kehittyessä.

Koska elektroniset kauppapaikat ja myös jotkin agenttijärjestelmät, esimerkiksi Aglets-järjestelmä, voivat toimia käyttöliittymänään WWW-sivu, voisi olla mahdollista toteuttaa vastaavat järjestelmät myös WAP-pohjaisen (Wireless Application Protocol) käyttöliittymän kautta. Koska WAP [WAP Forum, 1999] on tarkoitettu käytettäväksi lähinnä matkapuhelinten yhteydessä, voisi esimerkiksi elektronisten kauppapaikkojen käyttö olla mahdollista lähes mistä käsin tahansa.

Elektronisten kauppapaikkojen käyttö onnistuu kyllä nykyiselläänkin matkapuhelimen välityksellä. Elektronisia kauppapaikkoja voidaan käyttää lähes mistä käsin tahansa ottamalla kannettavasta mikrotietokoneesta Internet-yhteys matkapuhelimella tai käyttämällä matkapuhelimeen yhdistettyä PDA-laitetta (Personal Digital Assistant). WAP mahdollistaisi kuitenkin

riippumattomuuden kannettavasta tietokoneesta tai tavallista matkapuhelinta kalliimmasta yhdistetystä matkapuhelin-PDA-laitteesta.

Tulevaisuudessa elektronisia kauppapaikkoja voitaisiin käyttää matkapuhelimesta myös GPRS-yhteyden [Mobile Lifestreams Limited, 1999] (General Packet Radio Service) kautta. Tällä hetkellä langattomista viestimistä joudutaan muodostamaan palveluntarjoajaan yhteys, jossa yhteys on aktiivinen koko ajan. Koska agentteihin perustuvissa järjestelmissä agentit liikkuvat itsenäisesti, eikä niiden paluuaikaa voi välttämättä tarkasti ennustaa, voi agentteihin perustuvan palvelun käyttäminen ja agenttien paluun odottaminen tällaisella yhteydellä muodostua kalliiksi. GPRS mahdollistaa sen, että matkapuhelin on aktiivisessa yhteydessä palveluntarjoajaan vain silloin, kun palveluntarjoajan ja matkapuhelimen välillä siirretään tietoa. Tällainen paketteihin perustuva tiedonsiirto mahdollistaisi agenttien käyttämisen yhtälailla langattomista viestimistä kuin kiinteästi tietoverkkoon liitetyistä tietokoneista.

Nämä aiheet vaativat vielä lisää tutkimusta, jotta agenteilla toteutettujen elektronisten kauppapaikkojen turvallisuus ja luotettavuus saataisiin riittävän korkealle tasolle. Koska sekä agentit että elektroniset kauppapaikat ovat suurel- le käyttäjäkunnalle uusia asioita, on kynnys ottaa agentteihin perustuva elektroninen kauppapaikka jokapäiväiseen käyttöön verrattain korkea. Tällaiset järjestelmät pitäisikin saada erittäin luotettaviksi, jotta ne saavuttaisivat käyttäjien hyväksynnän ja luottamuksen ja voisivat näin ollen tulla yleisesti käyttöön. Uusilla ratkaisuilla voidaan tuottaa elektronisille kauppapaikoille lisäarvoa tai helpottaa kuluttajan asioimista elektronisissa kauppapaikoissa.

Useimpien ennusteiden mukaan elektronisen kaupan ja sitä myötä myös elektronisten kauppapaikkojen tulevaisuus näyttää loistavalta. Internet-liittymien jatkuvan kasvun myötä elektronisten kauppapaikkojen potentiaalinen asiakaskunta on koko ajan kasvamassa.

Ma [1999] kirjoittaa konsultointiyritys Forrester Researchin arvioineen, että Internetissä tapahtuvien elektronisten kauppojen arvo nousee vuoden 1997 14 miljardista dollarista 327 miljardiin dollariin viiden vuoden kuluessa. Maailmanpankin tutkimuksessa taasen on selvinnyt, että 80% Internetin tuottavasta tuote- tai palvelumarkkinoista on niin uusia, että niitä ei ollut vielä 12-18 kuukautta sitten.

Vaikka ennusteet lupaavatkin elektroniselle kaupankäynnille ruusuista tulevaisuutta, on syytä pitää mielessä, että usein elektronista kauppaa käy yritys, joka käy myös perinteistä kauppaa. Näin ollen elektronisen palvelun tarjoaminen on tällaiselle yritykselle ainakin aluksi enemmänkin yrityksen palvelujen saatavuuden parantamista kuin myynnin kasvattamista. Samalla yritys voi

kuitenkin valmistautua tulevaisuuden markkinoille ja hankkia tärkeää kokemusta elektronisen palvelun tarjoamisesta.

Pelkästään elektronisen kaupan kautta tuotteitaan tai palveluitaan tarjoava yritys on kilpailullisesti erilaisessa asemassa verrattuna yritykseen, joka kauppaa tuotteitaan sekä elektronisesti että perinteisesti. Hurjista kasvuennusteista huolimatta elektroninen kauppa on vielä tällä hetkellä niin pienimuotoista perinteiseen kauppaan verrattuna, että pelkästään elektronisen palvelun kautta tuotteitaan tarjoavalle yritykselle jokainen asiakas merkitsee yhtä asiakasta vähemmän perinteisen kaupan alueella.

8. Yhteenveto

Huolimatta liikkuvien agenttien tähän astisesta vähäisestä suosiosta esimerkiksi staattisiin agentteihin verrattuna, voidaan liikkuvia agentteja kuitenkin käyttää hyväksi tehtävissä, jotka vaativat usean agentin yhteistyötä. Yksi tällainen tehtävä on elektroninen kaupankäynti. Liikkuvia agentteja voidaan käyttää hyödyksi esimerkiksi elektronisten kauppapaikkojen toteuttamisessa. Liikkuvilla agenteilla pystytään ratkaisemaan joitakin elektronisiin kauppapaikkoihin liittyviä ongelmia, mutta kaikkiin ongelma-alueisiin liikkuvat agentit eivät tuo ratkaisua. Esimerkiksi ostosten maksaminen on pelkästään elektroniselle kaupankäynnille spesifinen ominaisuus, johon tällä hetkellä olemassa olevat liikkuvien agenttien kehittämiseen tarkoitetut kehitysympäristöt eivät tarjoa luontevaa ratkaisua.

Vaikka liikkuvilla agenteilla voidaankin ratkaista joitakin elektronisen kaupan ongelmia, tuovat liikkuvat agentit mukanaan myös niille ominaisia ongelmia. Esimerkiksi järjestelmien monimutkaistuminen entisestään luo lisää virhetilanteiden mahdollisuuksia ja näin ollen tuo ongelmia elektronisen kauppapaikan toteuttajalle. Suurin liikkuvien agenttien ongelma on kuitenkin yhteisen järjestelmän tarve. Käytettäessä liikkuvia agentteja elektronisen kauppapaikan toteuttamiseen pitäisi kaikilla järjestelmän käyttäjillä olla käytössään sama agenttijärjestelmä.

Verrattuna perinteisiin elektronisiin kauppapaikkoihin pystytään liikkuvien agenttien avulla elektronisten kauppapaikkojen käyttöä kuitenkin laajentamaan esimerkiksi matkapuhelimiin. Tulevaisuudessa käyttöönotettavat tiedonsiirtostandardit tukevatkin tällaista kehityksen suuntaa. Agentille voidaan tehdä WAP-pohjainen käyttöliittymä, jolloin elektroniseen kauppapaikkaan lähetettäviä agentteja voidaan luoda matkapuhelimesta käsin. Liikkuvien agenttien käyttö elektronisten kauppapaikkojen tai muiden järjestelmien toteuttamisessa voikin laajentaa järjestelmien käytettävyyttä ja muuttaa tämänhetkistä käsitystä elektronisesta kauppapaikasta WWW-pohjaisine käyttöliittymineen.

Tutkielman tuloksena toteutettu liikkuviin agentteihin perustuva elektroninen kauppapaikka tarjoaa käyttäjälle mahdollisuuden asettaa tavaroitaan myytäväksi elektroniseen kauppapaikkaan. Vaikka toteuttamisen aikana kohdattiin sekä liikkuviin agentteihin että elektronisiin kauppapaikkoihin liittyviä ongelmia, toteutettu kauppapaikka täyttää sille asetetut tavoitteet toteuttaa pelkästään liikkuviin agentteihin perustuva elektroninen kauppapaikka. Kauppapaikkaa voidaan käyttää kauppakumppaneiden tunnistamiseen käytettyjen

tavaroiden markkinoilla. Toteutettu kauppapaikka on kokeellinen järjestelmä ja vaatisi jatkokehitystä ennen soveltumista laajempaan käyttöön. Järjestelmää voitaisiin kehittää esimerkiksi lisäämällä siihen useiden kauppapaikkojen yhteistoiminnallisuutta.

Lähdeluettelo

- [Akateeminen Kirjakauppa, 1999] Akateeminen Kirjakauppa, 1999. <http://kauppa.akateeminen.com/>. Tarkistettu 26.10.1999.
- [Amazon, 1999] Amazon.com Inc, 1999. <http://www.amazon.com>. Tarkistettu 26.10.1999.
- [Anttila, 1999] Anttila, 1999. <http://www.anttila.fi/>. Tarkistettu 26.10.1999.
- [Baker, 1998] A. Baker, *JAFMAS, a Java-based agent framework for multi-agent systems*, 1998. <http://www.ececs.uc.edu/~abaker/JAFMAS/index.html>. Tarkistettu 26.10.1999.
- [Bits & Pixels, 1999] Bits & Pixels, *Intelligent Agent Library*, 1999. <http://www.bitpix.com/techspec.htm>. Tarkistettu 26.10.1999.
- [Boxman, 1999] Boxman, 1999. <http://www.boxman.fi/>. Tarkistettu 26.10.1999. Sivuja uudelleenorganisoidaan.
- [Chavez ja Maes, 1996] A. Chavez ja P. Maes, *Kasbah: An agent marketplace for buying and selling goods*, 1996. <http://ecommerce.media.mit.edu/papers/paam96.pdf>. Tarkistettu 26.10.1999.
- [Chavez et al, 1997] A. Chavez, D. Dreilinger, R. Guttman ja P. Maes, *A real-life experiment in creating an agent marketplace*, 1997. <http://ecommerce.media.mit.edu/papers/paam97.pdf>. Tarkistettu 26.10.1999.
- [Chouldhury et al, 1998] V. Chouldhury, K. Hartzel ja B. Konsynski, Uses and consequences of electronic markets: an empirical investigation in the aircraft parts industry, *MIS Quarterly*, **22** (4), 1998, 471-508.
- [Espoo, 1999] Espoo, *Sähköinen asiointi on pian arkipäivää*, 1999. <http://www.espoo.fi/asiointi/index.htm>. Tarkistettu 26.10.1999.
- [Frost, 1996] R. Frost, *Java Agent Template*, 1996. <http://cdr.stanford.edu/ABE/JavaAgent.html>. Tarkistettu 26.10.1999.
- [Guttman et al, 1998] R. Guttman, A. Moukas ja P. Maes, *Agent-mediated electronic commerce: a survey*, 1998. <http://ecommerce.media.mit.edu/papers/ker98.pdf>. Tarkistettu 26.10.1999.
- [Harrison et al, 1995] C. Harrison, D. Chess ja A. Kershbaum, *Mobile agents: are they a good idea?*, Research report, IBM T.J. Watson Research Center, 1995. <http://www.research.ibm.com/massive/mobag.ps>. Tarkistettu 26.10.1999.
- [Heecheol, 1998a] J. Heecheol, *JATLite overview*, 1998. <http://java.stanford.edu/JATLiteOverview.htm>. Tarkistettu 26.10.1999.

- [Heecheol, 1998b] J. Heecheol, *JATLite API reference*, 1998.
<http://java.stanford.edu/JATLiteDoc.htm>. Tarkistettu 26.10.1999.
- [IBM, 1997a] IBM, 1997. <http://www.tabican.ne.jp/>. Tarkistettu 26.10.1999.
- [IBM, 1997b] IBM, 1997. <http://www.trl.ibm.co.jp/aglets/emplac/emplac.html>. Tarkistettu 26.10.1999.
- [IBM, 1999a] IBM, *IBM Aglets software development kit*, 1999.
<http://www.trl.ibm.co.jp/aglets/>. Tarkistettu 26.10.1999.
- [IBM, 1999b] IBM, *Overview of IBM CommonRules 1.0 alpha release*, 1999.
<http://www.research.ibm.com/rules/commonrules-overview.html>.
 Tarkistettu 26.10.1999.
- [IKV++, 1998] IKV++, *Grasshopper - The agent platform*, 1998.
<http://www.ikv.de/products/grasshopper/index.html>. Tarkistettu 26.10.1999.
- [Lahti, 1999] J. Lahti, 1999: verkkoasioinnin vuosi?, *Internet*, 4B/99, Helsinki Media Company Oy, 1999, 45-47.
- [Lange ja Oshima, 1998] D. Lange ja M. Oshima, *Programming and deploying Java mobile agents with Aglets*, Addison Wesley Longman Inc, 1998.
- [Li ja Messerschmitt, 1996] W. Li ja D. Messerschmitt, *Components of the Java-To-Go infrastructure*, 1996. <http://ptolemy.eecs.berkeley.edu/dgm/javatools/java-to-go/to-go-components.html>. Tarkistettu 26.10.1999.
- [Lippupalvelu, 1999] Lippupalvelu, 1999. <http://www.lippupalvelu.fi/>.
 Tarkistettu 26.10.1999. Sivuja uudelleenorganisoidaan.
- [Ma, 1999] M. Ma, Agents in e-commerce, *Communications of the ACM*, **42** (3), 1999, 78-80.
- [Merita, 1999] Merita, *Solo-tori*, 1999. <http://www.merita.fi/solotori/>.
 Tarkistettu 26.10.1999.
- [MikroPC, 1999] *MikroPC 7/99*, Talentum Oyj, 1999, 10.
- [MIT, 1997] Massachusetts Institute of Technology, 1997.
<http://ecommerce.media.mit.edu/Kasbah/index.html>. Tarkistettu 26.10.1999.
- [MIT, 1999] Massachusetts Institute of Technology, *Market Maker*, 1999.
<http://maker.media.mit.edu/>. Tarkistettu 26.10.1999.
- [Mitsubishi, 1998] Mitsubishi Electric Information Technology Center America, *Mobile agent computing*, 1998. <http://www.meitca.com/HSL/Projects/Concordia/MobileAgentsWhitePaper.html>. Tarkistettu 26.10.1999.
- [Mobile Lifestreams Limited, 1999] Mobile Lifestreams Limited, 1999.
<http://www.mobilegprs.com/>. Tarkistettu 26.10.1999.

- [Nakamura ja Yamamoto, 1997] Y. Nakamura ja G. Yamamoto, *Aglets-based e-Marketplace: concept, architecture and applications*, Research Report RT-0253 IBM Research, 1997.
- [Oshima et al, 1998] M. Oshima, G. Karjoth ja K. Ono, *Aglets Specification 1.1 draft*, 1998. <http://www.trl.ibm.co.jp/aglets/spec11.html>, 1998. Tarkistettu 26.10.1999.
- [Peapod, 1999] Peapod, 1999. <http://www.peapod.com/>. Tarkistettu 26.10.1999.
- [Pfisterer, 1998] C. Pfisterer, *The Mole cookbook or how to program a Mole agent*, 1998. <http://www.informatik.uni-stuttgart.de/ipvr/vs/projekte/mole/docs/cookbook.html>. Tarkistettu: 26.10.1999.
- [Ruok@.net, 1999] Ruok@.net, 1999. <http://www.ruoka.net/>. Tarkistettu 26.10.1999.
- [Ruokavarasto, 1999] Ruokavarasto, 1999. <http://www.ruokavarasto.fi/>. Tarkistettu 26.10.1999.
- [Salminen, 1999] S. Salminen, *Tietokonemyyjät Netissä*, 1999. <http://www.digerati.fi/tietokonemyyjat/>. Tarkistettu 26.10.1999.
- [Sandholm, 1999] T. Sandholm, Automated negotiation, *Communications of the ACM*, **42** (3), 1999, 84-85.
- [Schubert et al, 1998] C. Schubert, R. Zarnekow ja W. Brenner, A methodology for classifying intelligent software agents, *Proceedings Of 6th European Conference on Information Systems*, 1998, 304-316.
- [SRI International, 1996] SRI International, *Open Agent Architecture*, 1996. <http://www.ai.sri.com/~oaa/>. Tarkistettu 26.10.1999.
- [Sun, 1997] Sun Microsystems Inc, *Object Serialization Specification*, 1997. <http://www.uta.fi/jdk/docs/guide/serialization/spec/serialTOC.doc.html>. Tarkistettu 26.10.1999.
- [Veho, 1999] Veho, 1999. <http://www.veho.fi/>. Tarkistettu 26.10.1999.
- [Venners, 1997] B. Venners, Under the hood: The architecture of aglets, *JavaWorld*, 1997, <http://www.javaworld.com/javaworld/jw-04-1997/jw-04-hood.html>. Tarkistettu 26.10.1999.
- [WAP Forum, 1999] Wireless Application Protocol Forum Ltd, 1999. <http://www.wapforum.org/>. Tarkistettu 26.10.1999.
- [Wong et al, 1999] D. Wong, N. Paciorek ja D. Moore, Java-based mobile agents, *Communications of the ACM*, **42** (3), 1999, 92-102.
- [Väestörekisterikeskus, 1999a] Väestörekisterikeskus, *Henkilön sähköinen tunnistaminen*, 1999. <http://www.vaestorekisterikeskus.fi/sahtun.htm>. Tarkistettu 26.10.1999.

- [Väestörekisterikeskus, 1999b] Väestörekisterikeskuksen lehdistötiedote 3.5.1999: *Sähköisen asioinnin varmenne- ja hakemistopalvelutoimittajat valittu*, 1999. <http://www.vaestorekisterikeskus.fi/tied9919.htm>. Tarkistettu 26.10.1999.
- [Ykköshalli, 1999] Ykköshalli, 1999. <http://nettikauppa.yhalli.k-supermarket.fi>. Tarkistettu 26.10.1999.